

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO INDUSTRIAL ELÉCTRICO

Título del proyecto:

RIEGO DE HUERTA AUTOMATIZADO POR
ARDUINO

MEMORIA

Alumno: David Bermúdez Alegre
Tutor: Vicente Senosiáin Miquélez
Pamplona, 12 de junio de 2014

Índice

1.	Introducción	6
1.1.	Objetivo del proyecto	6
1.2.	Justificación del proyecto	6
1.3.	¿Por qué este proyecto es diferente a los demás?	7
1.4.	Planificación del PFC.....	8
1.5.	Ubicación de la huerta	9
2.	INTRODUCCIÓN A LOS MICROCONTROLADORES.....	12
2.1.	¿Qué es un microcontrolador?	12
2.2.	Procesador.....	12
2.3.	Memoria no volátil	12
2.4.	Memoria de datos (RAM)	12
2.5.	Puertas de entrada y salida	13
2.6.	Placa de Arduino: LEONARDO	14
2.6.1.	Memoria.....	15
2.6.2.	Entradas y salidas	15
2.6.3.	Comunicación	16
2.6.4.	Programación	16
2.6.5.	Protección contra picos de corriente en el USB	17
2.6.6.	Características físicas.....	17
3.	ENTORNO DE DESARROLLO	19
3.1.	Estructura de un programa	20
3.2.	Variables	21
3.3.	Funciones	22
3.4.	Otras estructuras de control	23
3.4.1.	IF.....	23
3.4.2.	While.....	23
4.	SENSORES Y OTROS ELEMENTOS DEL PROYECTO	24
4.1.	Sensores de humedad	24
4.1.1.	Sensores de humedad capacitivos	24
4.1.2.	Sensor de humedad resistivo	26
4.1.3.	Sensor de humedad resistivo con comparadores de tensión	28
4.1.4.	Elección de los sensores.....	31

4.1.5.	Pruebas realizadas con el sensor de humedad.....	32
4.1.6.	Instalación del sensor	35
4.1.7.	Electrólisis.....	36
4.2.	Relé	37
4.3.	Sensores de nivel de líquidos.....	43
4.3.6.	Sensor de nivel de flotador.....	43
4.3.7.	Sensor de ultrasonidos.....	45
4.4.	Bombas de agua	48
4.5.	Electroválvula.....	50
4.6.	Depósitos de agua	51
4.7.	Led SMD de 3 colores	54
4.8.	Caja estanca	55
5.	RIEGO POR GOTEIO.....	56
5.1.	¿Qué es el riego por goteo?.....	56
5.2.	Historia	56
5.3.	Evolución del riego por goteo	57
5.4.	Ventajas del riego por goteo	58
5.5.	Inconvenientes del riego por goteo	59
5.6.	Tabla de riego de hortalizas	60
6.	DIAGRAMA DE FLUJO.....	62
7.	MONTAJE Y PUESTA EN MARCHA.....	63
7.1.	Montaje del sensor de humedad	63
7.2.	Ajustes el programa para Arduino	64
7.3.	Puesta en marcha	65
7.4.	Montaje Huerta.....	67
7.5.	Maqueta	67
8.	SINOPSIS DEL PROYECTO	68
8.1.	Propósito del proyecto	68
8.2.	Elección de materiales.....	68
8.2.6.	Arduino.....	68
8.2.7.	Sensores.....	68
8.2.8.	Bombas de agua	68
8.2.9.	Electroválvula	68

8.2.10.	Relé.....	69
8.2.11.	Sensores de nivel	69
8.2.12.	Cableado	69
8.3.	Montaje	69
8.4.	Otros aspectos a tener en cuenta.....	69
8.5.	Código de programación	70
9.	PRESUPUESTO	71
10.	PLANOS	72
10.1.	Plano maqueta	72
10.2.	Plano Huerta	73
11.	Código Arduino	74
11.1.	Código de programación para la maqueta	74
11.2.	Código de programación para la huerta	76

1. Introducción

1.1. Objetivo del proyecto

El presente Proyecto de final de carrera tiene como objetivo la realización del automatismo necesario para el control del riego de una huerta. La idea es realizar el control de una huerta de la forma más barata posible, utilizando una amplia variedad de materiales e incluso reutilizándolos. De esta manera sería un sistema que podría comprarlo casi cualquier persona, con poco presupuesto.

La idea de realizar el automatismo del riego de una huerta se debió a la necesidad de un familiar, que necesitaba ir cada fin de semana al pueblo para regar el huerto. Con la idea de automatizar el riego y de hacerlo de la manera más barata posible nace este proyecto.

Pensé en poder reciclar materiales de lavadoras o coches, motores de agua o electroválvulas, combinados con un Arduino que controlaría todos los elementos.

Para saber cuándo efectuar el riego, se pensó en medir la humedad del terreno mediante unos sensores existentes para Arduino que son extremadamente baratos (unos 3€).

El sistema de control electrónico está formado por dos partes: una parte está ubicada en el exterior, formada por el/los sensores, y la segunda en el interior o resguardado que consta del microcontrolador (Arduino) y los diferentes relés para los automatismos.

1.2. Justificación del proyecto

Por todos es sabido que los recursos naturales no deben desperdiciarse. Por este motivo el proyecto solo utiliza los recursos necesarios para un correcto crecimiento de las plantas. Con este sistema se consigue un ahorro de agua y electricidad, ya que se gestiona de manera automática y efectúa el riego solo cuando verdaderamente es necesario, todo ello mediante el/los sensor(es) de humedad colocado(s).

Desde el punto de vista académico se deseaba experimentar con el microcontrolador Arduino, concretamente con el LEONARDO. El controlador tiene un gran potencial para poder desarrollar este tipo de instalaciones y a un muy bajo coste. Otro aspecto interesante también es el tipo de programación que utiliza este controlador.

A la hora de elegir el proyecto, un aspecto importante es poder aumentar los conocimientos de programación y utilización de sensores.

1.3. ¿Por qué este proyecto es diferente a los demás?

Cuando empezamos a leer este proyecto, podemos pensar que es un proyecto más de control de riego por Arduino, que no aporta nada novedoso salvo que se cambien unos sensores por otros.

Este proyecto surge con una serie de premisas bien claras y que es lo que hace que este proyecto nazca.

La primera razón por la que se piensa en este proyecto es para realizar un control automatizado y de la manera más barata. Existen otros proyectos en los que el riego se controla por multitud de sensores que controlan la radiación solar, la humedad ambiente, la temperatura, la cantidad de agua caída,... Este proyecto tiene la intención de realizar el riego controlando la humedad de la tierra alrededor de la plantación, y así, con datos de la humedad en tiempo real y mediante una monitorización 24 horas al día poder controlar el riego. No es una simple colección de datos para luego decidir si se riega o no, es un monitoreo constante para ver la necesidad de agua en tiempo real.

Tal y como comentaba anteriormente, el sensor que utilizaremos es un sensor de humedad de tierra que se puede conseguir por menos de 3€ y que nos arroja unos buenísimos resultados, con una exactitud altísima, prácticamente parecida a los sensores de marcas conocidas que cuestan diez veces más.

Pero si por algo es un proyecto diferente, es porque el proyecto está pensado en sus inicios para ser barato pero también universal. Universal en el sentido de está pensado en un tipo de controladores como son los Arduino, con una serie de sensores, pero el resto de elementos para el riego pueden ser de lo más diversos. Podremos conectar casi cualquier sistema que deseemos, cualquier bomba, electroválvula o gotero que queramos, tengamos o podamos adquirir. Por tanto, incluso aunque dispongamos de un sistema de riego ya instalado, este control de riego podremos introducirlo en el sistema y él será el encargado de controlar el riego.

Por hacer un resumen podemos decir que se trata de un proyecto:

- 1- Nos ayudara a gestionar y ahorrar agua
- 2- Nos evitará tener que estar vigilándolo el huerto constantemente
- 3- Casi podremos adaptarlo a nuestro sistema de riego para automatizarlo, si disponemos de uno
- 4- Crear un sistema de riego reutilizando elementos
- 5- Crear un sistema de riego barato

1.4. Planificación del PFC

A continuación se detallarán los pasos o etapas necesarias para realizar el proyecto.

Para empezar, fue necesario ampliar los conocimientos de electrónica para decidir qué componentes serían necesarios emplear (por ejemplo saber qué sensores iba a necesitar), leer detalladamente las hojas de características de todos y adquirir conocimientos de cómo montar este tipo de circuitos.

La primera parte del proyecto, consistía en seleccionar un Arduino que controle todo el proyecto, buscar los sensores de humedad para Arduino y diferentes motores y electroválvulas para controlar el riego y por último sensores para controlar el agua de los tanques.

Una vez elegidos los componentes, se procedió al estudio de lo que sería la parte lógica del Arduino (el microcontrolador). Para ello fue necesario aprender sobre su funcionamiento, arquitectura interna y externa, y lenguaje de programación.

Por último, también realice un estudio del riego por goteo y de diferentes hortalizas, para saber cómo colocar adecuadamente el/los sensores de humedad y la cantidad de agua necesaria, para controlar el riego

Con los conceptos teóricos ya claros, se procedió al montaje de circuitos de prueba:

- 1- Prueba de diferentes sensores de humedad resistivos, para conocer los valores máximos y mínimos y el funcionamiento del módulo digital.
- 2- Pruebas con los distintos sensores de nivel para el agua.
- 3- Conexión y programa para el cálculo de distancias con el sensor de ultrasonidos.
- 4- Conexiones de las bombas de agua y electroválvulas empleadas para el riego.
- 5- Escribir el programa y probar que todo funciona correctamente.

1.5. Ubicación de la huerta

El proyecto que presento, lo situamos en una parcela de Arguedas. Arguedas es una villa y un municipio español de la Comunidad Foral de Navarra, situado en la merindad de Tudela, en la Ribera de Navarra y a 66,93 km de la capital de la comunidad, Pamplona. Su población en 2013 fue de 2335 habitantes.

Este pueblo está situado al norte de la Ribera del Ebro. La zona habitada está entre el inicio de las Bardenas Reales y la orilla norte del Ebro. Tiene zonas más hundidas que el Ebro, como sería el barrio de Venecia (nombre dado popularmente debido a las inundaciones sufridas).

Esta localidad que se caracteriza por ser una villa eminentemente agrícola y sus habitantes han vivido principalmente de los frutos de la tierra y del ganado.

Por las parcelas pasa un canal de regadío, el cual se abre levantando una compuerta manual.

Esta zona navarra consta de una climatología con precipitaciones más bien escasas y veranos cálidos y secos.

El clima de la Rivera navarra y por lo tanto de Arguedas es mediterráneo –continental y se caracteriza por los importantes contrastes climáticos entre el invierno y el verano, la escasez de precipitaciones y el viento.

La temperatura media anual es de 14°C y llueve entre 400 y 450 litros por m² al año en unos 60 días.

El viento más destacado en Arguedas es el de noroeste, llamado Cierzo, este viento determina la bonanza del clima, con Cierzo el verano es agradable y cuando sopla en invierno el ambiente es muy frío. El viento de sur, cálido y normalmente desagradable se denomina bochorno.

Dicha parcela está situada a 2.5 km al suroeste de Arguedas. Concretamente es esta que podemos ver rodeada de rojo



En la parte de arriba tiene un camino, y justo alado del camino pasa el canal de riego. El canal se comunica con la huerta por la parte de abajo del camino abriendo una paradera manualmente.



A continuación unas fotos de lo que actualmente se encuentra en la huerta



2. INTRODUCCIÓN A LOS MICROCONTROLADORES

Al tratarse del cerebro del riego de la huerta, creí oportuno dedicar un capítulo a explicar el funcionamiento de los microcontroladores y en especial el modelo que he utilizado para realizar este proyecto.

En la actualidad, los aparatos electrónicos están presentes en todos los aspectos de la vida diaria y su presencia sigue aumentando día tras día. Un motivo que ha impulsado este fenómeno es el desarrollo de los microcontroladores, ya que gracias a ellos resulta sencillo y económico crear aplicaciones para las que hasta hace poco era necesarios grandes y complejos sistemas electrónicos.

2.1. ¿Qué es un microcontrolador?

Recibe el nombre de controlador, el dispositivo que se emplea para el gobierno de uno o varios procesos. Es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: Memoria, CPU y Entradas/Salidas.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo, su implementación física ha variado frecuentemente. Hace tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores, que se rodeaban con chips de memoria y E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo, pero completo computador integrado en un sólo encapsulado.

Todos los microcontroladores están formados por cuatro elementos básicos.

2.2. Procesador

Es el “cerebro” del microcontrolador. Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso. Su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

2.3. Memoria no volátil

Sería el equivalente al disco duro de un ordenador. Se utiliza para almacenar las instrucciones del programa. Puede ser ROM, EPROM, OTP, EEPROM o FLASH, dependiendo de las necesidades: si es necesario programarlo varias veces.

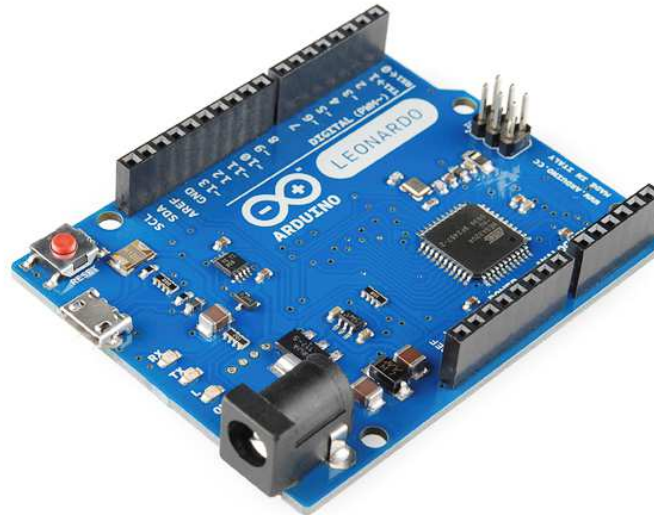
2.4. Memoria de datos (RAM)

Se utiliza para almacenar las variables y los cambios de información que se produzcan en el transcurso del programa. Por lo tanto es necesaria una memoria de lectura y escritura y que además soporte muchos ciclos de trabajo. La memoria RAM estática (SRAM) es la más utilizada, aunque en algunos casos también se utiliza memoria EEPROM para poder almacenar datos una vez finalizado el programa.

2.5. Puertas de entrada y salida

Este elemento permite al microprocesador comunicarse con el mundo exterior. A través de las patas del circuito integrado, puede emitir y recibir señales por medio de corrientes eléctricas. Dependiendo de marcas y modelos, estas señales pueden ser analógicas o digitales, en serie o en paralelo. Mediante estas entradas y salidas se pueden efectuar muchísimas funciones, leer información de sensores, activar o desactivar diferentes mecanismos,...

2.6. Placa de Arduino: LEONARDO



El microcontrolador que vamos a utilizar en este proyecto es el Arduino. Se trata de unas placas open hardware por lo que su diseño es libre de distribución y utilización.

En este caso, tenemos una placa LEONARDO, la cual es de la propia marca Arduino, y está basada en un microcontrolador ATmega32u4. 20 pines de Entrada/Salida, todas ellas configurables como digitales. 7 de ellas con capacidad PWM. 12 pueden ser utilizadas como entradas analógicas con una resolución de 10 bits. Funcionan a 16MHz, tiene una conexión Micro-Usb y un conector de alimentación.

Contiene todo lo necesario para apoyar el microcontrolador, simplemente conectar a un ordenador con un cable USB y/o la clavija de alimentación con un adaptador de CA a CC para funcionar.

Una de las novedades que incorpora el Leonardo, es que se diferencia de las placas anteriores en que el **ATmega32u4** incorpora comunicación USB, eliminando la necesidad de un procesador secundario. Esto permite que Arduino Leonardo maneje nuestro teclado o nuestro ratón, por ejemplo. Ya existen librerías en la página oficial de Arduino para llevar a cabo esto que acabo de mencionar.

La placa Arduino Leonardo, puede ser alimentada a través de la conexión micro USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

Para usar alimentación externa (no mediante USB), ésta puede venir de un adaptador de CA a CC o de una batería. El adaptador (fuente de alimentación), lo podemos conectar mediante un jack de 2.1mm con centro positivo a la toma de alimentación de la placa. Sin embargo, los cables de los que disponga nuestra batería, los podemos insertar en los pines GND y Vin de la placa.

La placa puede operar mediante una alimentación externa de 6 a 20 Voltios. En caso de alimentar la placa con menos de 7 Voltios, el pin de 5 Voltios puede no llegar a este valor y la placa podría volverse inestable. Sin embargo, si utilizamos más de 12 Voltios, el regulador de tensión puede sobrecalentarse y llegar a dañar nuestra placa. Por lo tanto, el rango recomendado de alimentación es de 7 a 12 Voltios.

2.6.1. Memoria

El ATmega32u4 que incorpora el Arduino Leonardo, posee 32 KB (4 KB utilizados para el bootloader). También ofrece 2,5 KB de SRAM y 1 KB de EEPROM (los cuales pueden ser leídos y escritos con la librería EEPROM).

2.6.2. Entradas y salidas

Cada uno de los 20 pines digitales de entrada/salida que posee la placa, se pueden utilizar mediante las funciones de siempre: `pinMode()`, `digitalWrite()` o `digitalRead()`.

Cada uno de los 20 pines de E/S digitales se pueden utilizar como entrada o salida, usando las funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Cada pin nos proporcionará un máximo de 40 mA y, tampoco podrá recibir más de este valor.

Serial: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos serie TTL con la capacidad de hardware serie del ATmega32U4. Hay que tener en cuenta que, en Arduino Leonardo, la clase Serial se refiere a la comunicación USB (CDC) ; para serie TTL en los pines 0 y 1, utilizaremos la clase Serial1.

IST: 2 (SDA) y 3 (SCL). Soportan la comunicación TWI utilizando la librería Wire.

Interrupciones externas: 2 y 3. Estos pines se pueden configurar para desencadenar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Ver `attachInterrupt ()` para obtener más detalles.

PWM: 3, 5, 6, 9, 10, 11 y 13. Proporcionan 8-bit de salida PWM con la función `analogWrite ()`.

SPI: en la cabecera de ICSP. Estos pines soportan la comunicación SPI utilizando la librería SPI. Debemos tener en cuenta que los pines de SPI no están conectados a cualquiera de los pines digitales de entrada/salida, como están en Arduino UNO. Sólo están disponibles en el conector ICSP.

LED: 13. Hay un LED smd conectado al pin digital 13. Cuando el pin tiene valor alto, el LED está encendido, cuando se pasa a valor bajo, se apaga.

Entradas Analógicas: A0-A5, A6 – A11 (en los pines digitales 4, 6, 8, 9, 10, y 12). La placa Arduino Leonardo tiene 12 entradas analógicas, las cuales van desde A0 hasta A11. También pueden ser utilizadas como entradas/salidas digitales. Los pines A0-A5 están situados en el mismo lugar que en Arduino UNO; las entradas A6-A11 se corresponden con las entradas/salidas digitales de los pines 4, 6, 8, 9, 10 y 12, respectivamente. Cada entrada analógica puede proporcionar 10 bits de resolución, es decir 1024 valores diferentes (desde 0 hasta 1023). Por defecto, la medida de las entradas analógicas van desde 0 a 5 Voltios, aunque se puede cambiar el extremo superior de su rango con el pin AREF y la función `analogReference()`.

AREF. Tensión de referencia para las entradas analógicas. Se utiliza con `analogReference()`.

Reset: reinicia el microcontrolador.

También puedes ver el mapeo entre los pines de Arduino y los puertos del ATmega32u4.

2.6.3. Comunicación

La placa Arduino Leonardo ofrece importantes facilidades para obtener comunicación con un ordenador, otro Arduino o incluso con otros microcontroladores. El ATmega32u4 incorpora comunicación serie UART TTL (5Voltios) disponible en los pines digitales 0 (RX) y 1 (TX), los cuales se han comentado anteriormente.

Este microcontrolador también nos permitirá establecer una comunicación serial a través de USB (CDC) y aparecer como puerto COM virtual con en nuestro sistema operativo. El integrado también actúa a modo de dispositivo de máxima velocidad USB 2.0, utilizando los drivers estándar de COM USB. Para Windows, se necesitará un archivo con extensión .inf.

El IDE de Arduino incluye un monitor de puerto serie que nos va a proporcionar la opción de transmitir texto desde la placa Arduino y, hacia la misma. Ya sabemos que los LEDs TX y RX de la placa parpadearán cuando se transmitan datos a través de la conexión USB al ordenador (pero no para la comunicación de serie en los pines 0 y 1).

Para establecer una comunicación serie en cualquiera de los pines digitales de Arduino, usaremos la librería `SoftwareSerial`

El ATmega32u4 también es compatible con la comunicación I2C (TWI) y SPI. El IDE de Arduino incluye la librería `Wire` (enlace compartido anteriormente) para simplificar el uso del bus I2C, puedes consultar la información para obtener más detalles sobre esto. La librería SPI la utilizaremos para la comunicación SPI. Mediante Arduino Leonardo, podemos emular un teclado y un ratón genérico, y puede ser programado para controlar estos dispositivos de entrada utilizando clases `Keyboard` & `Mouse`.

2.6.4. Programación

La placa Arduino Leonardo se programará con el software de Arduino y, tan sólo deberemos elegir el modelo de placa que vamos a utilizar, elegir el puerto en el que hemos instalado el Arduino (virtual port, de un USB a un COM).

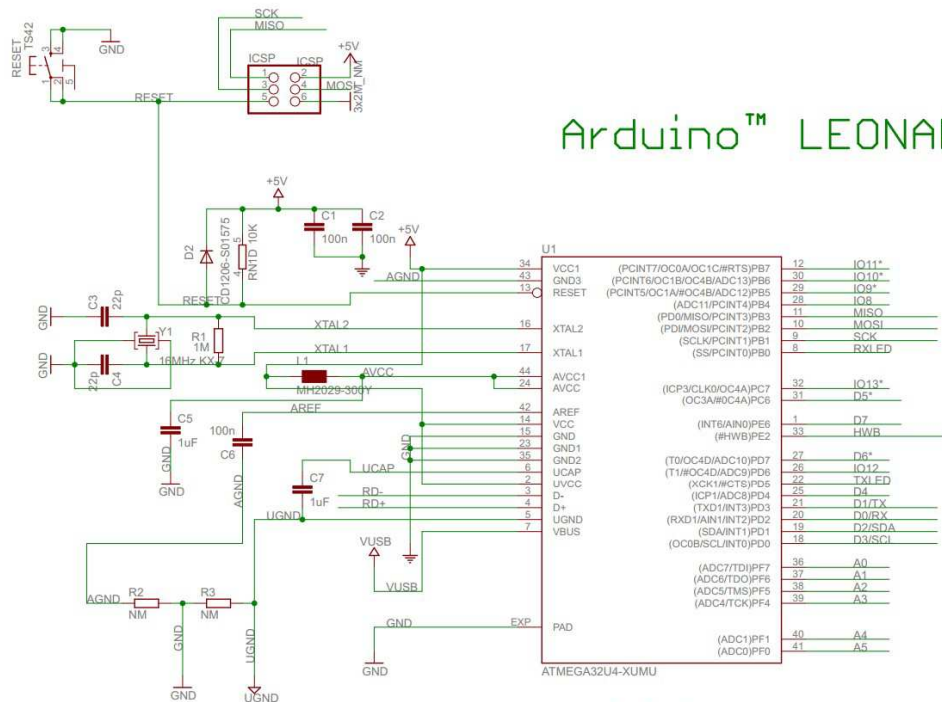
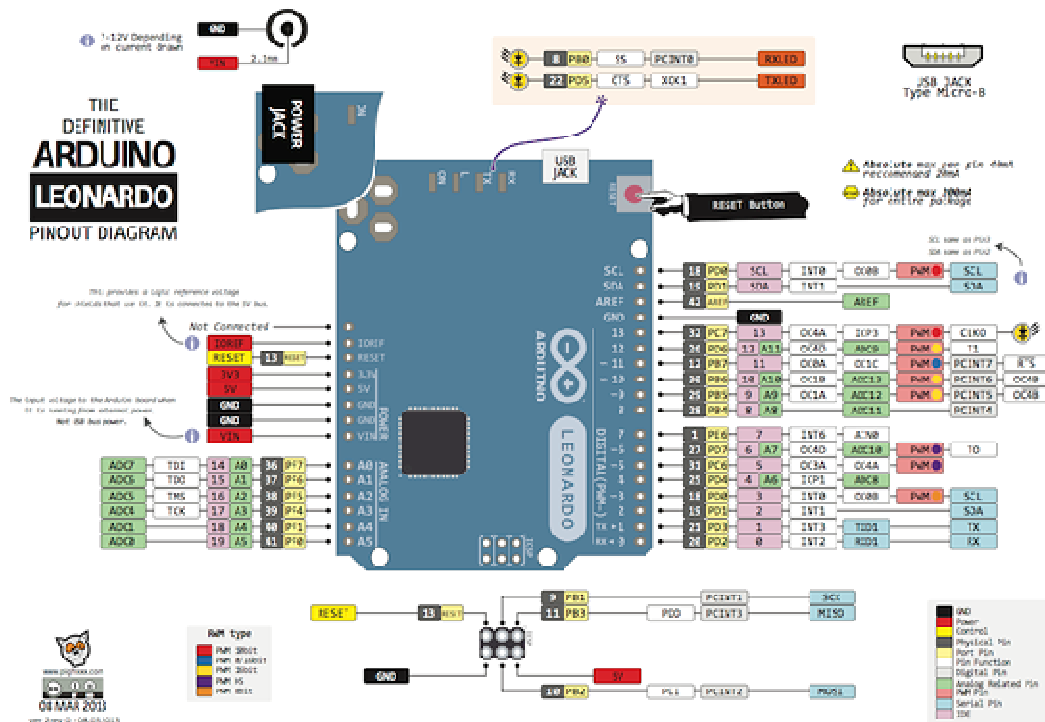
2.6.5. Protección contra picos de corriente en el USB

La placa Arduino Leonardo posee un polifusible reseteable que protege los puertos USB de nuestro ordenador de las variaciones de corriente. Aunque, si no todos, la mayoría de los ordenadores ofrecen una protección interna, el fusible nos proporciona una protección extra. Si aplicamos más de 500 mA al puerto USB del ordenador, el fusible automáticamente corta la conexión hasta que el cortocircuito o sobrecarga desaparezca.

2.6.6. Características físicas

La longitud máxima y la anchura de la placa Arduino Leonardo son 2,7 y 2,1 pulgadas respectivamente, con el conector USB y el conector de alimentación aumenta la dimensión anterior. Posee cuatro orificios para los tornillos que nos permiten sujetar la placa a una superficie o caja. La distancia entre los pines digitales 7 y 8 es de 160 milésimas de pulgada (0,16') y no 100 milésimas como en el resto de pines. Así obtendremos algo más de espacio para nuestros cables.

A continuación unos esquemas tanto del chip **ATmega32u4** como de la propia placa Leonardo



3. ENTORNO DE DESARROLLO

Los programas hechos con Arduino se dividen en tres partes principales: *estructura*, *valores* (variables y constantes), y *funciones*. El Lenguaje de programación Arduino se basa en C/C++.

Este lenguaje de programación tiene su origen en el año 1970, se creó a partir de un lenguaje anterior denominado BCPL. Se popularizó rápidamente debido a que resultó ser altamente compatible, pero surgió el problema de que se crearon numerosas implementaciones. Para solucionar este inconveniente se creó un estándar ANSI para el

Lenguaje C, que fue adoptado finalmente en 1989 y es el que actualmente se utiliza. El C se considera un lenguaje de computadora de nivel medio. Esto significa que está a medio camino entre un lenguaje de alto nivel, que resulta más sencillo de interpretar, y el lenguaje ensamblador, que es más funcional. Este lenguaje tiene la capacidad de manejar bits, bytes y direcciones aunque esto no le impide ser muy portable, esto es que se adapta fácilmente a cualquier tipo de computador. Se puede considerar un lenguaje estructurado ya que tiene la capacidad de crear subrutinas o funciones de forma que los eventos que se producen dentro de las mismas no causan efectos colaterales en otras partes del programa. Se dice que este es un lenguaje para programadores ya que apenas tiene restricciones, y es muy versátil. Inicialmente se utilizó C para programación de sistemas y este es un motivo muy importante por el que este lenguaje se ha extendido tanto entre los programadores.

Para programar la placa es necesario descargar de la página web de Arduino el entorno de desarrollo (IDE). Se dispone de versiones para Windows y Mac, así como las fuentes para compilarlas en LINUX.

3.1. Estructura de un programa

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones.

```
void setup(){  
  estamentos;  
}  
void loop(){  
  estamentos;  
}
```

setup() es la parte encargada de recoger la configuración y **loop()** es la que contiene el programa que se ejecutará cíclicamente (de ahí el termino loop –bucle-). Ambas funciones son necesarias para que el programa trabaje.

La función de configuración debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta sólo una vez, y se utiliza para configurar o inicializar pinMode (modo de trabajo de las E/S), configuración de la comunicación en serie y otras.

La función bucle (loop) siguiente contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc) Esta función es el núcleo de todos los programas de Arduino, la cual realiza la mayor parte del trabajo.

Setup()

La función setup() se establece cuando se inicia un programa -sketch. Se emplea para iniciar variables, establecer el estado de los pins, inicializar librerías, etc. Esta función se ejecutará una única vez después de que se conecte la placa Arduino a la fuente de alimentación, o cuando se pulse el botón de reinicio de la placa.

```
void setup()  
{  
  pinMode(pin, OUTPUT); // configura el 'pin' como salida  
}
```

Después de llamar a **setup()**, la función **loop()** hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa esté respondiendo continuamente ante los eventos que se produzcan.

```
void loop()  
{  
  
}
```


3.2. Variables

Una variable es una forma de nombrar y almacenar un valor para su uso posterior por el programa, tal como datos de un sensor o un valor intermedio utilizado en un cálculo. Las variables en Arduino.

Antes de que sean usadas, todas las variables deben ser declaradas. Declarar una variable significa definir su tipo y, opcionalmente, establecer el valor inicial (inicializando la variable). Las variables no tienen que ser inicializadas (asignándole un valor) cuando son declaradas.

Las variables en el lenguaje de programación C, usado por Arduino, tienen una propiedad llamada ámbito. Al contrario de lo que pasa en lenguajes como BASIC en los que todas las variables son globales.

Una variable global puede verse por todas las funciones de un programa. Las variables locales son sólo visibles desde la función en las que están declaradas. En el entorno Arduino, cualquier variable declarada fuera de la función (por ejemplo `setup()`, `loop()`, etc.), es una variable global.

Cuando los programas crecen en tamaño y complejidad, las variables locales son una forma útil de asegurar que sólo una función tiene acceso a sus propias variables. Esto previene errores de programación al evitar que una función modifique, involuntariamente, variables de otras funciones.

A veces es útil declarar e inicializar una variable dentro de un bucle `for`. Esto crea una variable que sólo se puede acceder desde dentro del bucle.

Algunos tipos de variables:

- `int`: Integers (Números enteros) son el principal tipo de datos para almacenar números, y guardan valores de 2 bytes. Esto produce un rango entre -32,768 hasta 32,767 (valor mínimo de -2^{15} y un valor máximo de $(2^{15}) - 1$).
- `long`: Las variables de tipo Long son variables de tamaño extendido para almacenamiento de números, y 32 bits (4 bytes), desde -2,147,483,648 hasta 2,147,483,647.
- `char`
- `byte`
- `unsigned int`
- `unsigned long`
- `float`
- `double`

Para el programa que desarrollare solo empleare las variables INT y LONG.

3.3. Funciones

Segmentar el código en funciones permite al programador crear piezas modulares de código que realizan una tarea definida y vuelven a la zona del programa en la que fueron llamadas. El caso típico para crear una función es cuando uno necesita realizar la misma acción múltiples veces dentro de un mismo programa.

La estandarización de fragmentos de código en funciones tiene diversas ventajas:

- Las funciones ayudan al programador a ser organizado. Además ayudan a conceptualizar el programa.
- Las funciones codifican una acción en un lugar, así que sólo deben ser depuradas de errores una vez.
- Reducen las posibilidades de error en modificaciones, si el código debe ser cambiado.
- Las funciones hacen el sketch más pequeño y más compacto por que las secciones de código se reutilizan varias veces.
- Hacen más fácil la reutilización de código en otros programas por hacerlo más modular y, como efecto paralelo, usando funciones se obtiene un código más legible.

3.4. Otras estructuras de control

Para el control del programa del Arduino disponemos de muchas estructuras para poder realizar cualquier tipo de función que deseemos y manejar todo tipo de sensores. Las funciones más utilizadas de las que disponemos son IF (comparador si-entonces), IF....ELSE (comparador si...sino), FOR (bucle con contador), SWITCH CASE (comparador múltiple), WHILE (bucle por comparación booleana), DO...WHILE

3.4.1. IF

if, el cual puede ser usado en conjunto con uno o más operadores de comparación, comprueba si cierta condición se cumple, por ejemplo, si un input posee un valor mayor a cierto número. El formato para una comprobación *if* es el siguiente:

```
if (comparación)
{
    // Hacer algo aquí.
}

x == y (x es igual a y)
x != y (x no es igual a y)
x < y (x es menor a y)
x > y (x es mayor a y)
x <= y (x es menor o igual a y)
x >= y (x es mayor o igual a y)
```

3.4.2. While

Los bucles *while* se ejecutan continuamente, hasta que la expresión de dentro del paréntesis, (), pasa a ser falsa. Algo debe modificar la variable comprobada, el bucle *while* nunca terminará. Lo que modifique la variable puede estar en el código, como una variable que se incrementa, o ser una condición externa, como el valor que da un sensor.

La sintaxis es la siguiente:

```
while(expresión)
{
    // Sentencia(s)
}
```

4. SENSORES Y OTROS ELEMENTOS DEL PROYECTO

En este apartado analizaremos diferentes tipos de sensores y elementos que podríamos utilizar para realizar este proyecto. Con sus pros y contras decidiremos cuales escogeremos. Este proyecto quiere hacer uso de los sensores y elementos más baratos posibles e incluso reutilizar elementos que podamos encontrar para realizar el riego, todo ello sin olvidar la fiabilidad de todos los elementos que escojamos.

El riego queremos controlarlo mediante la placa Arduino, pero este elemento necesita recibir la información de la humedad de la tierra para saber cómo debe de actuar.

4.1. Sensores de humedad

Existen en el mercado varios tipos de sensores de humedad y muchas marcas que comercializan dichos sensores. Básicamente podemos encontrar dos tipos de tecnologías en este tipo de sensores, capacitivos y resistivos.

Cada tipo de sensor tiene sus puntos fuerte e inconvenientes.

4.1.1. Sensores de humedad capacitivos

Este tipo de sensores calculan la humedad de la tierra, calculando el volumen de agua que se encuentra en el suelo mediante una técnica capacitiva.

Mediante cargas y descargas rápidas de los electrodos positivos y tierra (condensador) en el suelo, se genera un campo electromagnético cuya carga tiempo (t) está relacionada con la capacitancia (C) del suelo por donde R es la resistencia

$$t = R C \ln\left[\frac{V - V_f}{V_i - V_f}\right] \quad (1)$$

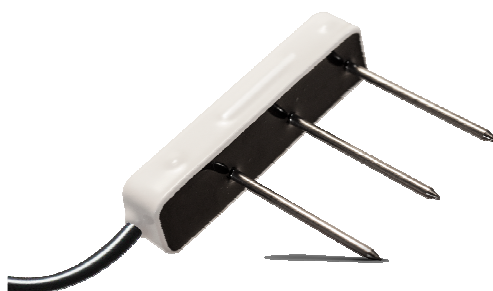
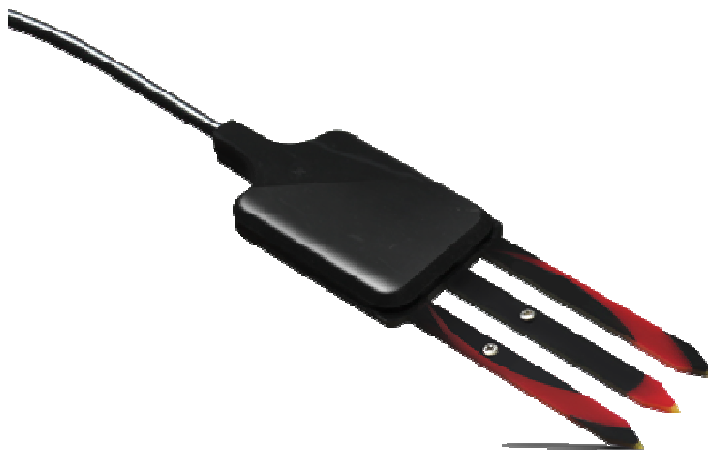
en serie, V es el voltaje en el tiempo t , V_i es el voltaje inicial y V_f es el voltaje aplicado. Además, para un condensador con factor geométrico F , la capacidad está relacionada con la permeabilidad dieléctrica (ϵ) del medio entre los electrodos del condensador, donde ϵ_0 es la permeabilidad del vacío.

$$C = \epsilon_0 \epsilon F \quad (2)$$

Por lo tanto, la permeabilidad del suelo puede ser determinada midiendo el tiempo de cambio (t) de un sensor enterrado en el suelo. En consecuencia, ya que el agua tiene una permeabilidad dieléctrica que es mucho mayor que los minerales del suelo o del aire, el tiempo t de carga en el suelo de la ecuación (1) puede correlacionarse con el contenido volumétrico de agua del suelo.

Existen versión de sensores capacitivos, que aparte de realizar lecturas de humedad en la tierra, pueden informarnos de la temperatura de la misma e incluso de la conductividad eléctrica.

Este tipo de sensores los descarté por su elevado precio.



4.1.2. Sensor de humedad resistivo



Este sensor de humedad puede leer la cantidad de humedad presente en el suelo que lo rodea. Es un sensor de baja tecnología, pero es ideal para el seguimiento de un jardín urbano, huerto o el nivel de agua de su planta de casa. Esta es una buena herramienta para tener controlado el jardín.

Este sensor utiliza las dos sondas para pasar corriente a través del suelo, y luego se lee la resistencia para conseguir el nivel de humedad. Más agua hace que el suelo conduzca más fácilmente la electricidad (menos resistencia), mientras que el suelo seco conduce la electricidad peor (mayor resistencia).

Especificaciones:

Fuente de alimentación: 3.3v a 5v

Señal de tensión de salida: 0 - 4.2v

Corriente: 35mA

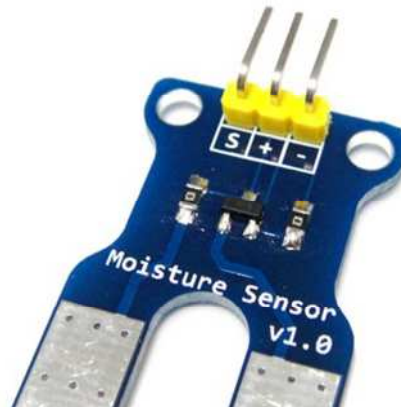
Pines:

1 - salida analógica (S)

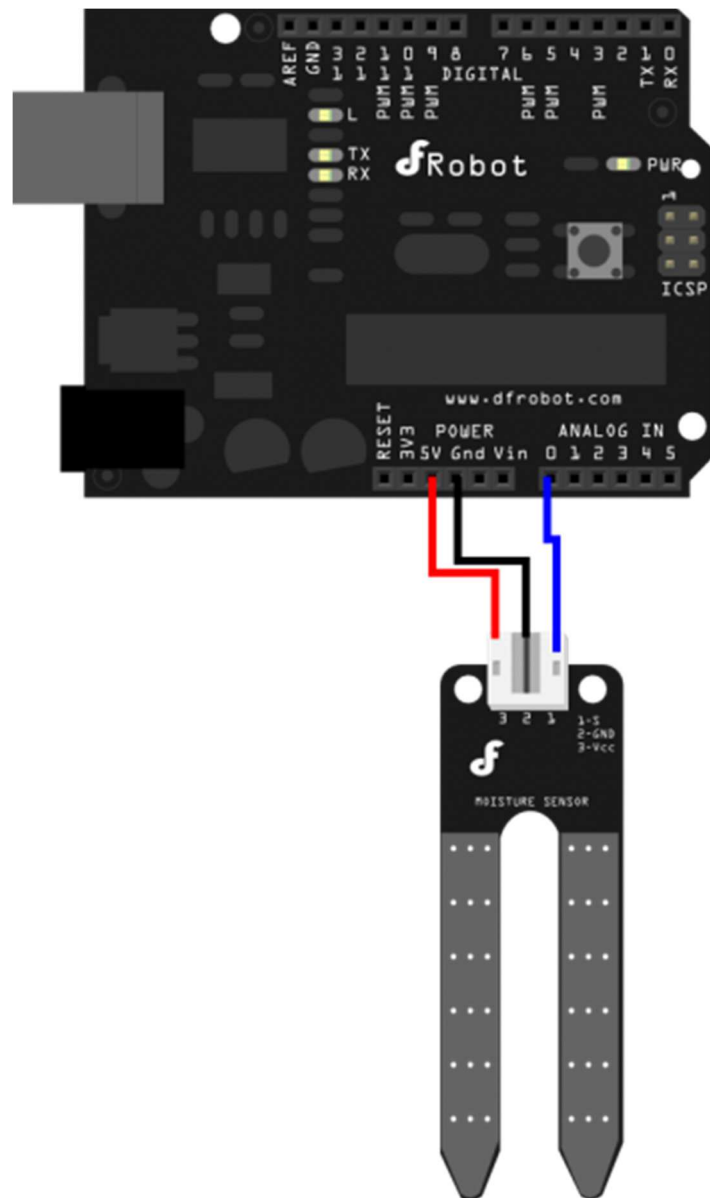
2 - GND (-)

3 - Power (+)

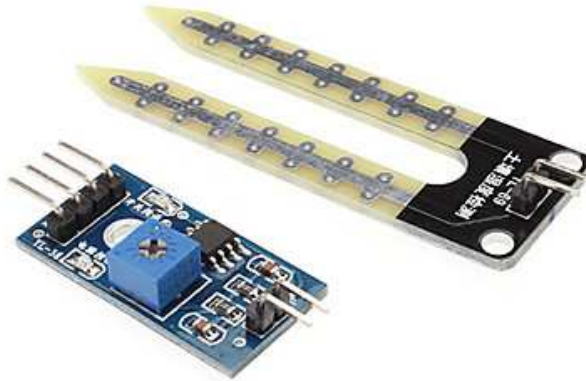
Tamaño: 60x20x5mm



La conexión la realizaremos de la siguiente manera:



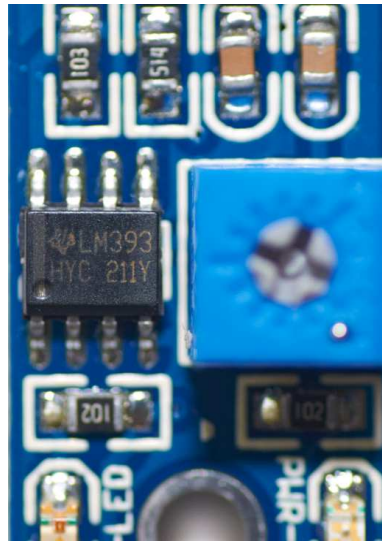
4.1.3. Sensor de humedad resistivo con comparadores de tensión



Este sensor de humedad es prácticamente igual que el anterior, pero incluye una placa que se coloca entre el Arduino y el sensor de humedad. Esta placa proporciona la posibilidad de hacer lecturas tanto analógicas (igual que el anterior sensor), como lecturas digitales.

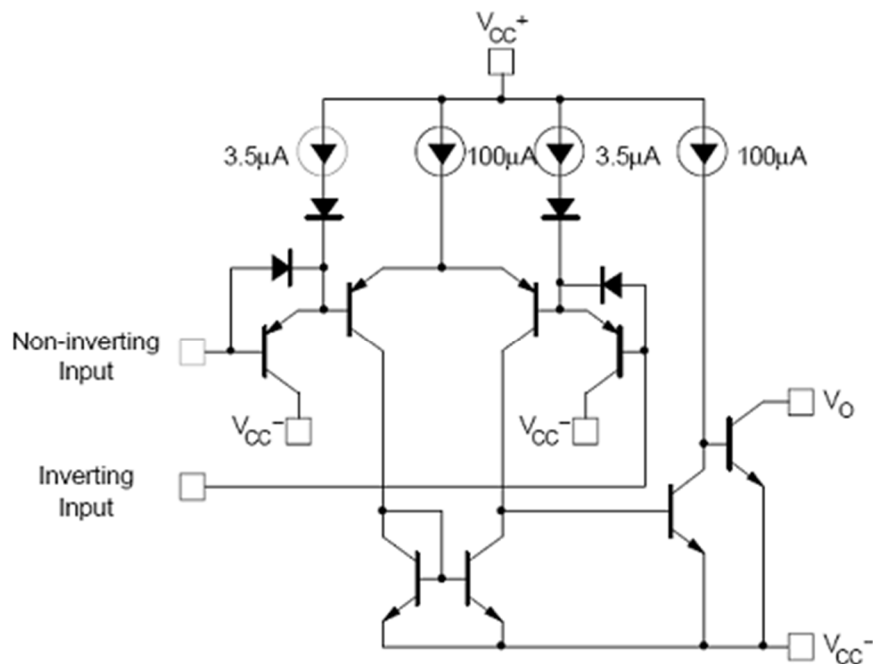
Las lecturas digitales consisten en devolver 0 o 1 al Arduino mediante el ajuste del potenciómetro azul que dispone la placa. Con ese potenciómetro ajustamos el punto de humedad en el que queremos que nos devuelva el comparador un 1. Si el valor es más bajo del punto en el que tenemos ajustado el comparador nos devolverá un 1, por tanto menos humedad de la que deseamos, si el punto de humedad se encuentra por encima del valor que hemos establecido nos devuelve un 0. Con estos valores realizaremos el programa para realizar el riego o las acciones que creamos oportunas.

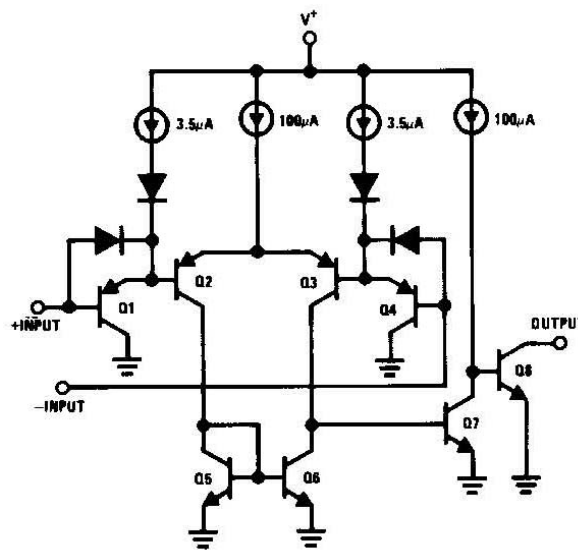
El circuito usado es el LM393. Este integrado dispone de dos comparadores en su interior.



Entre sus principales características encontramos que se puede alimentar con fuente doble ± 15 o simple de $+5$ V. Además posee una salida a colector abierto lo que permite seleccionar los niveles de la tensión de salida.

Si observamos cómo está construido vemos que tiene las entradas en bases abiertas y la salida en colector abierto.





Ganancia de tensión: 200 V/mV

Tiempo de respuesta: 1,3 μ seg.

Máxima tensión de entrada diferencial: 36V

Margen de tensión de entrada en modo común: 0 a $V_{cc} - 2$ V

Tensión Offset máximo: 9 mV


Corriente de entrada máxima: 400 nA

Corriente Offset máxima: 150 nA

Tensión de alimentación: Simple de 2 a 36 V, Doble de ± 1 V a ± 18 V

4.1.4. Elección de los sensores

Los sensores de tipo capacitivos decidí descartarlos por su elevado precio, ya que pueden costar 10 veces más que los resistivos. Los sensores resistivos podemos conseguirlos por unos 3€ tanto el sensor resistivo simple como la versión con el comparador. A continuación dos capturas donde adquirí los sensores.



Sensor de Humedad del Suelo KEYES contenido de Arduino Productos - Plata + Red
 ★★★★★ (3 reviews) SKU: 225594 (Añadido el 07/07/2013)

Precio: € **2,84**

Envío: el envío gratis A SPAIN

Despachado: Se enviará en los siguientes 7-10 días laborables

Directo: Worldwide EU

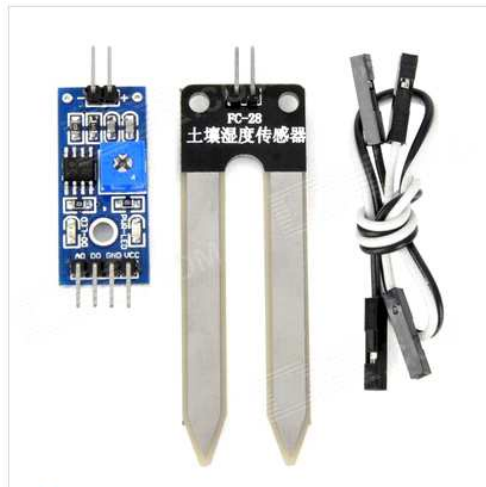
Cantidad:

AÑADIR A LA CESTA Añadir a la lista de deseos

Precio Ajustado Satisfacción 100% garantizada

Notificar un Error

Detalles del Producto
Reviews
Discusiones



Humedad Humedad detección del módulo del sensor del suelo - Azul + Negro + Blanco
 ★★★★★ (0 reviews) SKU: 200142 (Añadido el 10/04/2013)

Precio: € **2,19** **28% DESCUENTO** Lista de precios: €3,93

[> Más Opciones](#)

Envío: el envío gratis A SPAIN

Despachado: Se enviará en los siguientes 7-10 días laborables

Directo: Worldwide EU

Cantidad:

AÑADIR A LA CESTA Añadir a la lista de deseos

Precio Ajustado Satisfacción 100% garantizada

Notificar un Error

Detalles del Producto
Reviews
Discusiones

4.1.5. Pruebas realizadas con el sensor de humedad

Vuelvo a reiterar, que me decanté por comprar los sensores resistivos, ya que los capacitivos cuestan diez veces más que los resistivos.

Por tanto realicé un pedido de sensores resistivos, concretamente el que incluye el comparador. Cuando recibí los sensores resistivos lo primero que hice fue mirarme el datasheet de los sensores y sus especificaciones para comprender el funcionamiento del sensor y saber cómo realizar el conexionado.

Una vez tenía todo claro, realicé el siguiente programa para realizar mediciones en analógico y poder sacar los límites del sensor, tanto mínimo como máximo.

```
void setup()
{
  Serial.begin(57600);
}
void loop()
{
  Serial.print("Moisture Sensor Value:");
  Serial.println( (1023 - analogRead(0)));
  delay(1000);
}
```

Para las mediciones prepare una pequeña maceta con tierra lo más seca posible. Al principio, cuando el sensor estaba fuera de la maceta marcaba 0. Una vez introduje el sensor empezó a marcarme un valor de 147, así que empecé a echarle agua y pude comprobar como subía hasta unos 1008, en ese momento la tierra estaba encharcada. Por lo tanto el mínimo está en 0 y el máximo rondando los 1010. Estos valores son algo que podíamos suponer, ya que al tratarse de una lectura analógica y tener 10bits de resolución tenemos 1024 valores (de 0 a 1023).

Realicé la comparación de valores de humedad con un sensor de humedad comercial que me prestó un familiar, para comprobar si las mediciones eran correctas y fiables. Realicé una serie de mediciones con ambos sensores, y en los dos casos los resultados eran muy parejos. Cuando el sensor de Arduino arrojaba un valor de aproximadamente 450, el sensor comercial situaba su valor en el medio de su escala (5).

Con eso pude comprobar que el sensor de Arduino media correctamente y arrojaba valores correctos.

Por tanto podemos decir que el 0 corresponde al 0% de humedad en la tierra y el 1000 aprox es el 100% de humedad.

También pude comprobar como al echar agua poco a poco la humedad subía de manera progresiva y lineal, sin dar saltos en las mediciones.

Con todo esto y junto con unas especificaciones que me aportó uno de los vendedores de los sensores podemos afirmar el siguiente rango de valores para este tipo de sensores.

Rango de valores:

0 - 300: tierra seca

300 - 700: tierra húmeda

700 - 1020: tierra muy húmeda

Otra de las pruebas que realicé fue la de colocar el sensor de humedad con un cable a 5 metros de distancia. Para ello utilicé un cable de red de par trenzado de 8 hilos, ya que con este cable puedo llevar hasta 8 hilos, lo que me permitirá conectar más de un sensor con un solo cable. Las mediciones que realicé arrojaron una caída de tensión en los 5m de 0.03v. Con este valor de caída de tensión podemos asegurar que el funcionamiento de sensor será correcto y podemos conectar tranquilamente los sensores.

Un punto conflictivo para la instalación de los sensores es el aislamiento de las conexiones y de los diferentes elementos, pequeñas resistencias y demás, que llevan tanto el propio sensor de humedad como la placa. Para aislar los diferentes elementos se me ocurrió utilizar pegamento termofusible o sino también con silicona.

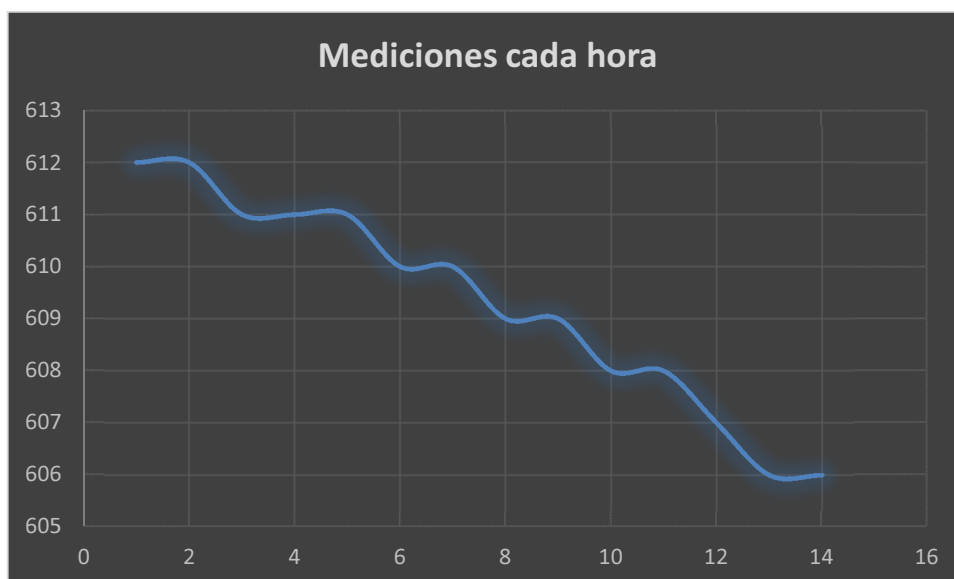
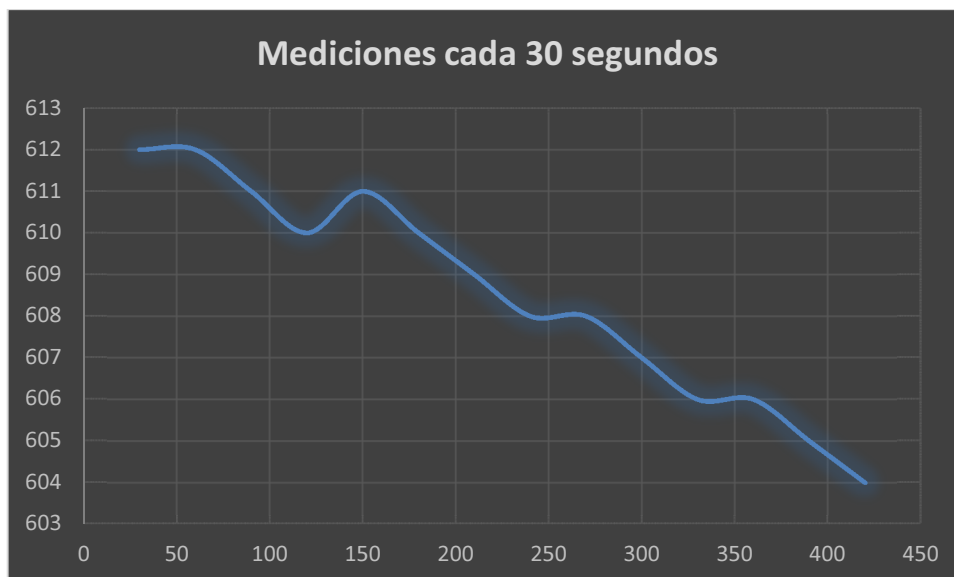
Yo realice una prueba con el pegamento termofusible en el sensor de humedad y funciono perfectamente. Le soldé el cable de red y después con el pegamento termofusible aislé todo perfectamente. Para probar la estanqueidad, sumergí el sensor en un vaso de agua y las lecturas fueron correctas, en ningún momento funcionó de una manera errática ni se produjo un cortocircuito ni nada por el estilo.

Por tanto, con estas pruebas realizadas podemos asegurar que el sensor de humedad se puede colocar a 5m del Arduino sin ningún tipo de problema y aguantaría cualquier tipo de condición a la que se le someta.



En cuanto a la exactitud de las lecturas se refiere, la utilización de 2 sensores en la misma zona efectuando mediciones, nos ofrecería una medición mucho más exacta. Los sensores se colocarían alrededor de la hortaliza que tengamos plantada, realizando una media de las lecturas para tener una medición más fiable.

Otro aspecto a tener en cuanto en las mediciones fue la frecuencia con la que se deben realizar las mediciones. Las pruebas que se realizaron arrojaron unos resultados clarísimos. Las mediciones no pueden realizarse con mucha frecuencia, y mucha frecuencia me refiero a mediciones con pocos segundos de intervalo de una medición a otra, ya que la tierra situada entre las dos puntas del sensor puede empezar a “secarse” y parecer que la humedad de la tierra está bajando.



Se observa que en el recipiente pequeño las mediciones cada poco tiempo son incorrectas. Realice mediciones cada 30 segundos y otras cada hora. Las mediciones cada 30 segundos arrojaban datos incorrectos.

Por tanto, pude comprobar como el sensor con el circuito comparador se muestra tremendamente exacto. Si lo usamos de manera analógica las lecturas son muy precisas, tanto utilizando con poco cable como con hasta 8 metros. Utilizando el modo digital, deberemos ajustar el potenciómetro en el punto en que deseemos que nos salte la señal de falta de humedad. Para los dos casos necesitaremos programas de Arduino diferentes.

Para ajustar la medición digital, lo que hice fue poner el sensor con el modo analógico, dejé la tierra con la humedad mínima que deseaba tener siempre, y conecté el sensor en modo digital ajustando el potenciómetro a ese punto de humedad. A partir de ese momento, cuando llegase a esa humedad mínima, el circuito manda una señal al Arduino y saltaría el riego.

Para realizar siempre unas lecturas correctas es recomendable cambiar el sensor cada 4-6 meses. Su coste es ridículo y supondría menos de 6€ al año.

4.1.6. Instalación del sensor

Al seleccionar un lugar para la instalación, es importante recordar que el suelo adyacente a la superficie del sensor, tiene una fuerte influencia en la lectura del sensor. Por lo tanto, la instalación del sensor requiere que se haga con especial cuidado y dedicación para que las lecturas sean lo más precisas posibles. Es posible que los palos, corteza, raíces u otros materiales atrapados entre el sensor afecten negativamente a las lecturas. También es importante poner el sensor cerca de la zona de goteo, ya que sino las lecturas no serán correctas y se producirá un riego más abundante de lo necesario. Por último, la instalación del sensor se debe de realizar de manera vertical, teniendo en cuenta la profundidad de las raíces del cultivo, por tanto si es necesario enterrar el sensor se excavara y enterrara.

Se debe tener especial cuidado al insertar los sensores en el suelo denso, los contactos pueden romperse si se ejerce una fuerza excesiva.



4.1.7. Electrólisis

El mayor problema que muestran los sensores de humedad de suelo resistivos, es su tendencia a oxidarse. Este fenómeno se conoce como electrólisis, y hace que el sensor deje de medir correctamente tomando medidas erróneas.

La electrólisis es el proceso que separa los elementos de un compuesto por medio de la electricidad. En ella ocurre la captura de electrones por los cationes en el cátodo (una reducción) y la liberación de electrones por los aniones en el ánodo (una oxidación).

Este proceso fue descubierto accidentalmente en 1800 por William Nicholson mientras estudiaba el funcionamiento de las baterías. Entre los años 1833 y 1836 el físico y químico inglés Michael Faraday desarrolló las leyes de la electrólisis que llevan su nombre y acuñó los términos.

El proceso de la electrólisis sería el siguiente:

- Se aplica una corriente eléctrica continua mediante un par de electrodos conectados a una fuente de alimentación eléctrica y sumergida en la disolución. El electrodo conectado al polo positivo se conoce como ánodo, y el conectado al negativo como cátodo.
- Cada electrodo atrae a los iones de carga opuesta. Así, los iones negativos, o aniones, son atraídos y se desplazan hacia el ánodo (electrodo positivo), mientras que los iones positivos, o cationes, son atraídos y se desplazan hacia el cátodo (electrodo negativo).
- La manera más fácil de recordar toda esta terminología es fijándose en la raíz griega de las palabras. Odos significa camino. Electrodo es el camino por el que van los electrones. Catha significa hacia abajo (catacumba, catástrofe). Cátodo es el camino por donde caen los electrones. Anas significa hacia arriba. Ánodo es el camino por el que ascienden los electrones. Ion significa caminante. Anión se dirige al ánodo y catión se dirige al cátodo. La nomenclatura se utiliza también en pilas. Una forma fácil también de recordar la terminología es teniendo en cuenta la primera letra de cada electrodo y asociarla al proceso que en él ocurre; es decir: en el ánodo se produce la oxidación (las dos palabras empiezan con vocales) y en el cátodo la reducción (las dos palabras comienzan con consonantes).
- La energía necesaria para separar a los iones e incrementar su concentración en los electrodos es aportada por la fuente de alimentación eléctrica.
- En los electrodos se produce una transferencia de electrones entre estos y los iones, produciéndose nuevas sustancias. Los iones negativos o aniones ceden electrones al ánodo (+) y los iones positivos o cationes toman electrones del cátodo (-).
- En definitiva lo que ocurre es una reacción de oxidación-reducción, donde la fuente de alimentación eléctrica se encarga de aportar la energía necesaria.

Por este motivo, coloqué un relé entre la fuente de 5v del Arduino y el sensor, para alimentarlo solo cuando realicemos las lecturas y después desconectarlo.

4.2. Relé

Para la activación del sensor de humedad, de las diferentes bombas de agua y/o electroválvulas utilizaremos unos relés de la marca SONGLE, montados sobre unas placas con contactos específicamente diseñadas para su utilización con Arduino. Son unos relés de unas dimensiones bastantes reducidas.

Lo bueno de estos relés es que tiene un precio muy reducido y ocupan muy poco lugar, por lo que facilitaran la instalación al requerir de poco espacio.

Ahora unas fotos del relé puesto en la placa para Arduino



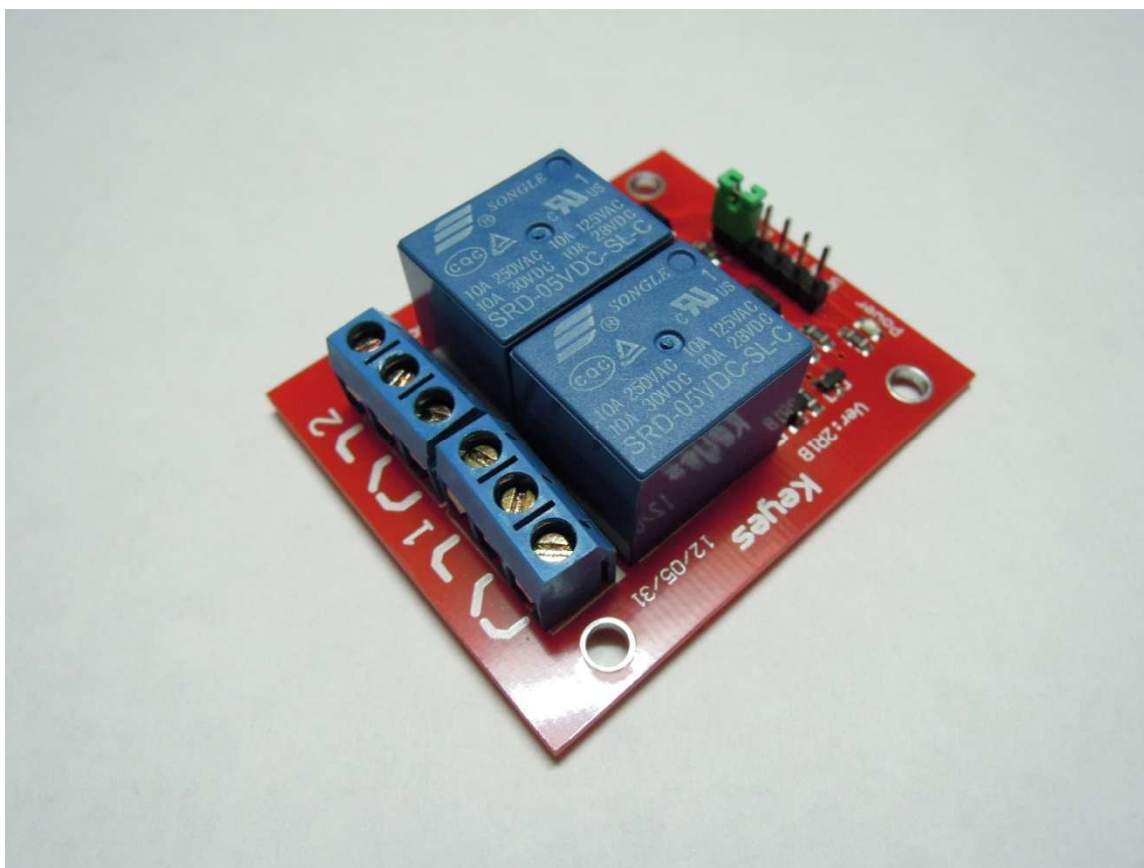
Foto de la conexión a Arduino



Foto del conexionado



También existen modelos de dos o más relés (4, 8 o 16)



Conexiones para Arduino




Foto del conexionado para los motores



A continuación la hoja de características del relé

SONGLE RELAY

	RELAY ISO9002	SRD
---	---------------	------------



1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for high density P.C. board mounting technique.
- UL,CUL,TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.
 (Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

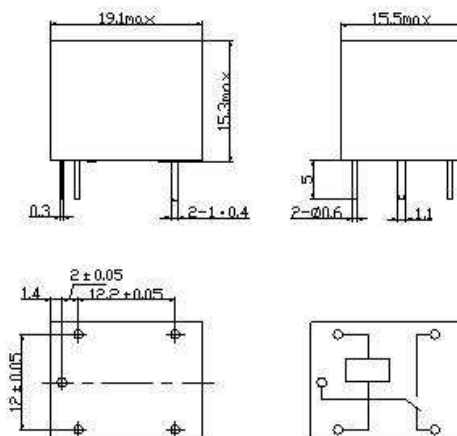
3. ORDERING INFORMATION

SRD	XX VDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil sensitivity	Contact form
SRD	03、05、06、09、12、24、48VDC	S:Sealed type	L:0.36W	A:1 form A
		F:Flux free type	D:0.45W	B:1 form B C:1 form C

4. RATING

CCC FILE NUMBER: CQC03001003731 10A/250VDC
 UL/CUL FILE NUMBER: E167998 10A/125VAC 28VDC
 TUV FILE NUMBER: R50056114 10A/250VAC 30VDC

5. DIMENSION (unit:mm) DRILLING (unit:mm) WIRING DIAGRAM



6. COIL DATA CHART (AT20°C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) $\pm 10\%$	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75%Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
	48	48	7.5	6400				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	89.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				
	48	48	10	4500	abt. 0.51W			

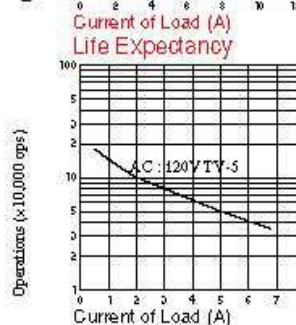
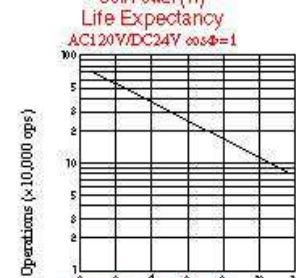
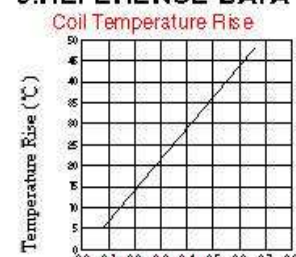
7. CONTACT RATING

Item	Type	SRD	
		FORM C	FORM A
Contact Capacity Resistive Load ($\cos\Phi=1$)		10A 125VAC	10A 30VDC 10A 250VAC
Inductive Load ($\cos\Phi=0.4$ L/R=7msec)		3A 120VAC 3A 28VDC	5A 120VAC 5A 28VDC
Max. Allowable Voltage		250VAC/110VDC	250VAC/110VDC
Max. Allowable Power Force		800VAC/240W	1200VA/300W
Contact Material		AgCdO	AgCdO

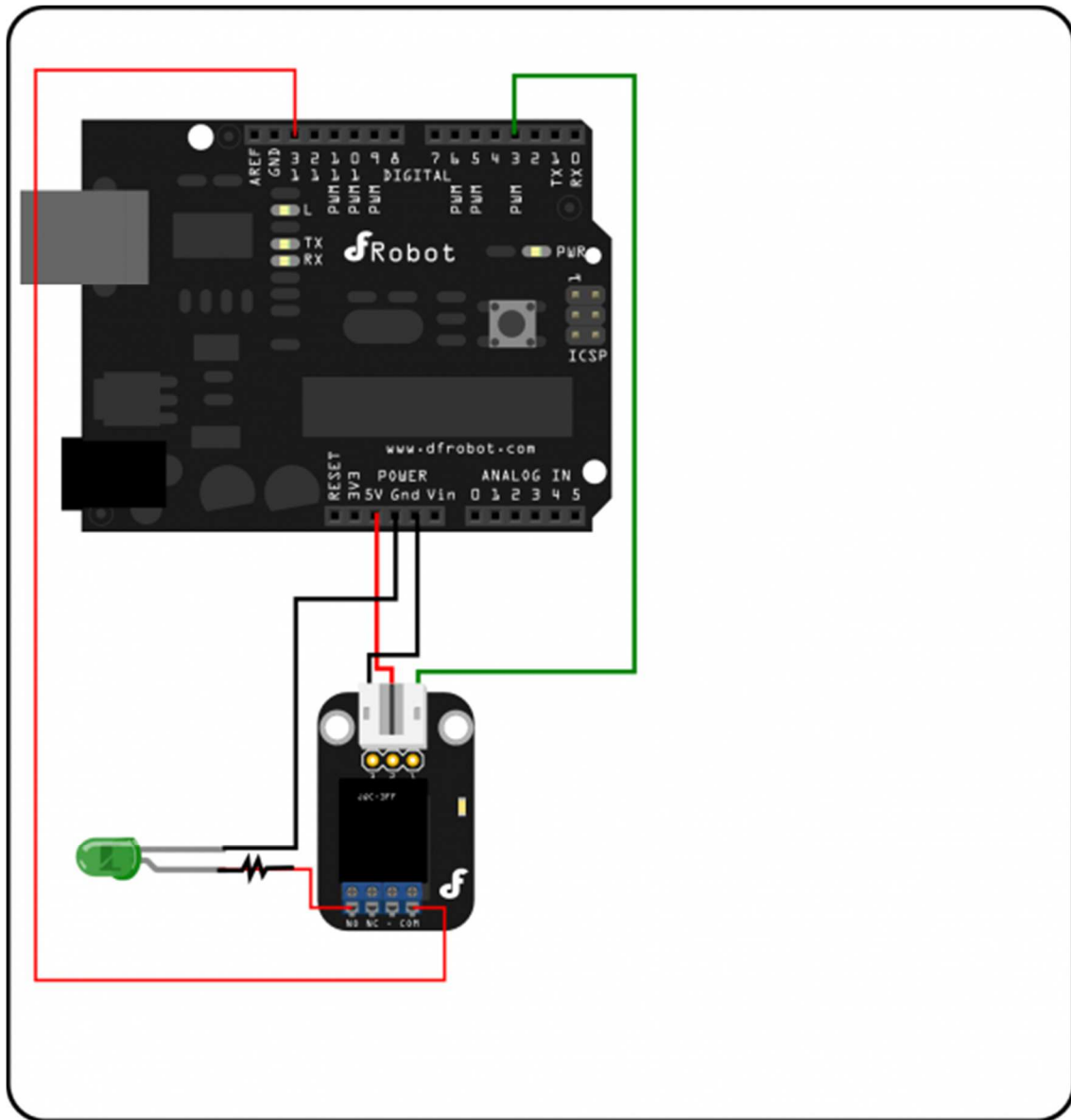
8. PERFORMANCE (at initial value)

Item	Type	SRD
Contact Resistance		100m Ω Max.
Operation Time		10msec Max.
Release Time		5msec Max.
Dielectric Strength		
Between coil & contact		1500VAC 50/60HZ (1 minute)
Between contacts		1000VAC 50/60HZ (1 minute)
Insulation Resistance		100 M Ω Min. (500VDC)
Max. ON/OFF Switching		
Mechanically		300 operation/min
Electrically		30 operation/min
Ambient Temperature		-40°C to +85°C
Operating Humidity		45 to 85% RH
Vibration		
Endurance		10 to 55Hz Double Amplitude 1.5mm
Error Operation		10 to 55Hz Double Amplitude 1.5mm
Shock		
Endurance		100G Min.
Error Operation		10G Min.
Life Expectancy		
Mechanically		10 ⁷ operations. Min. (no load)
Electrically		10 ⁵ operations. Min. (at rated coil voltage)
Weight		abt. 10grs.

9. REFERENCE DATA



La conexión del relé la realizaríamos de la siguiente manera. En este caso sería para encender un led.



4.3. Sensores de nivel de líquidos

Los sensores de nivel de líquidos controlarán la altura mínima y máxima del depósito, activan y desactivando la bomba de llenado según sea necesario. Tras estudiar las diferentes opciones, decidí que el sensor por ultrasonidos me daría todos los datos necesarios en el tanque del agua y encima en tiempo real, aparte de facilitarme la programación. Para ellos será necesario calibrar el sensor en el tanque en el que se utilice, ajustando los valores mínimo y máximo donde se desee. La medición de la altura del agua en el tanque se realiza por ultrasonidos, por tanto se calcula la distancia a la que se encuentra el agua del sensor. y por tanto midiendo una primera vez a diferente alturas, podemos saber la altura del agua en cada momento y de esta manera decidir llenar el depósito del agua cuando deseemos, parándolo cuando llegue a una determinada altura.

4.3.6. Sensor de nivel de flotador

Al principio del proyecto, cuando se pensó en las diferentes maneras de controlar el agua del tanque, se pensó en poder hacerlo con sensores de nivel de líquidos más rudimentarios como estos.



Estos sensores de tipo flotador son sensores para montaje en el interior de depósitos. Fabricado con polipropileno, son aptos para agua y líquidos similares. Cuando el flotador magnético llega a la posición más alejada de la tuerca cerrera su contacto. La sujeción al depósito se realiza mediante rosca.

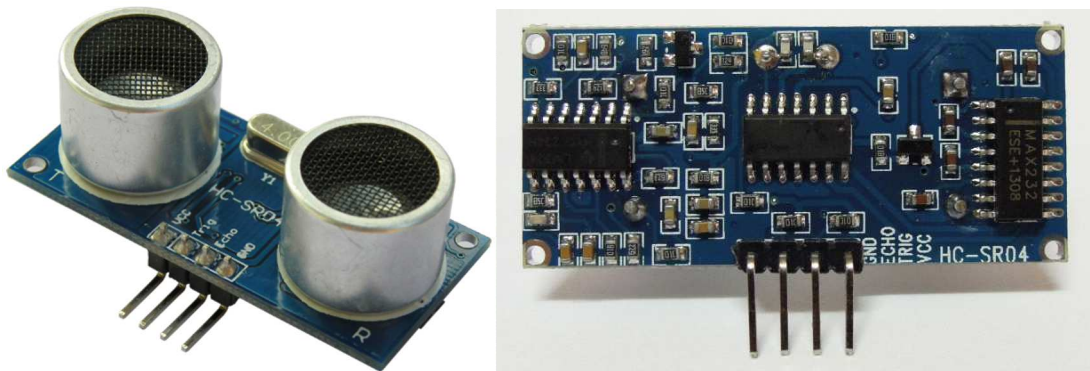
Características:

- Cable de 35cm
- Voltaje Máximo 100 VDC
- Corriente 100 mA

Este tipo de sensores se pueden encontrar fácilmente para Arduino, pero tienen unos inconvenientes de instalación, y es que para instalarlos tienes que hacer agujeros en los laterales del bidón. Otro inconveniente es que si se desea cambiar la altura del sensor se tiene que volver a agujerear.

Por estos inconvenientes decidí buscar otra opción para realizar el control del nivel del agua.

4.3.7. Sensor de ultrasonidos



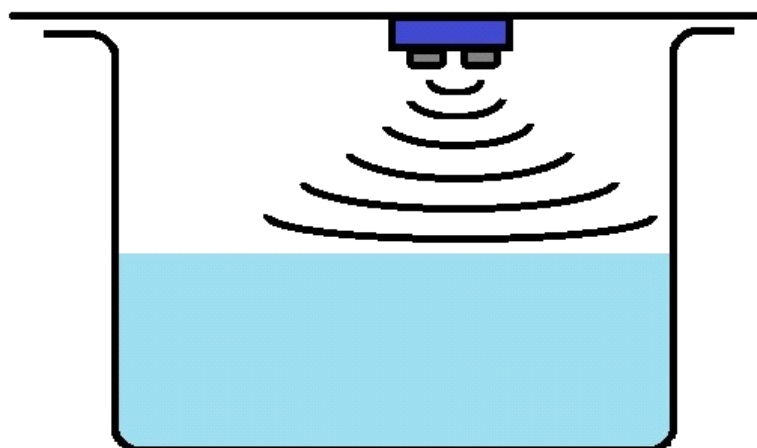
El HC-SR04 es un sensor de ultrasonidos de bajo coste que no sólo puede detectar si un objeto está presente, sino que también puede detectar la distancia a la que se encuentra dicho objeto.

El sensor de distancia ultrasónico HC-SR04 tiene la capacidad de medir la distancia por medio de la diferencia en tiempo entre la transmisión y recepción de una serie de pulsos que el modulo envía y captura. Este sensor es capaz de medir con precisión distancias de entre 2cm y 400cm de manera no invasiva o sin contacto.

El sensor de ultrasonidos nos permite controlar de una manera más exacta la altura del agua dentro del depósito, y podremos ajustar los valores de nivel máximo y nivel mínimo modificando el programa de Arduino y no teniendo que mover los sensores y agujereando el bidón cada vez que deseemos cambiar las alturas. Por esta serie de ventajas decidí decantarme por la utilización de este sensor de ultrasonidos.

La única cosa que tendremos que tener cuidado a la hora de instalar el sensor es proteger el sensor del agua de lluvia o que se moje. Si se monta dentro de bidón y se cubre bien el hueco por donde pasen los cables y el propio sensor, es suficiente.

La instalación del sensor dentro del tanque/bidón la realizaremos de la siguiente manera.



Básicamente el modulo incluye el transmisor y receptor de ultrasonidos, y el circuito de control. Ofrece una excelente detección sin contacto con elevada precisión y lecturas estables. La velocidad del sonido en el aire (a una temperatura de 20°C) es de 343m/s. Por cada grado centígrado que sube la temperatura, la velocidad del sonido aumenta en 0.6 m/s.

Tensión de trabajo	DC 5V
Corriente de trabajo	15mA
Frecuencia de trabajo	40HZ
Detección MAX	4m
Detección mínima	2cm
Angulo de medición	<15º
Resolución	La precisión puede variar entre los 3mm o 0.3cm
Dimensiones	45mm x 20mm x 15mm

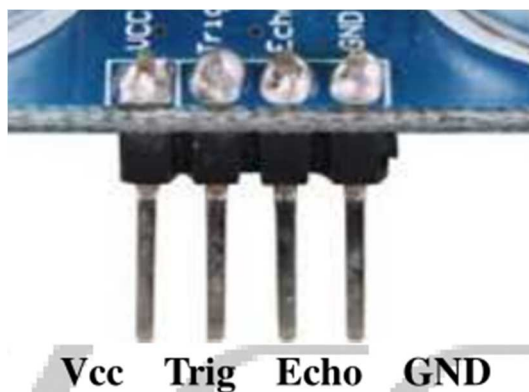
Las conexiones del sensor son las siguientes:

VCC: Alimentación +5V (4.5V min – 5.5V max)

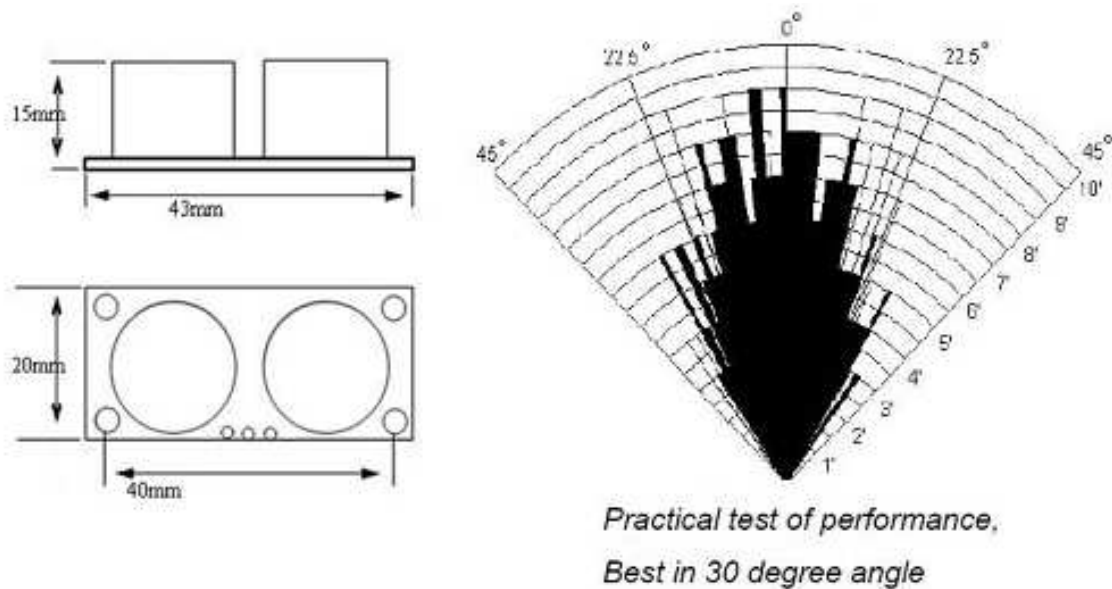
TRIG: Trigger entrada (input) del sensor (TTL)

ECHO: Salida (output) Echo del sensor (TTL)

GND

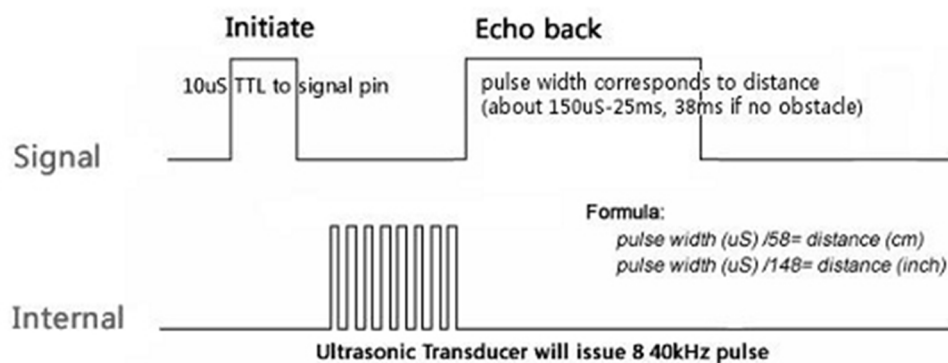


Las dimensiones del sensor y un test de funcionamiento:



Para medir la distancia con el HC-SR04 hay que generar un pulso en el pin Trig de un ancho o tiempo de 10µs como mínimo. Al mismo tiempo hay que monitorizar la señal que llega al pin de Echo. La distancia calculada por el sensor se corresponde a la fórmula:

$$\text{Distancia} = (\text{Ancho de Pulso} \cdot \text{Velocidad Sonido}) / 2$$



4.4. Bombas de agua

Algo a destacar en este proyecto es la polivalencia de poder emplear cualquier material que queramos comprar o podamos encontrar, no es necesario tener que hacer el montaje con una marca determinada (con el gasto de dinero que ello conlleva el trabajar con sistemas exclusivos de una marca en concreto). En el apartado de bombas de agua podemos encontrar en el mercado una amplia variedad de bombas según nuestras necesidades, con un abanico de precios muy amplio. También tendremos la posibilidad de conseguir bombas de agua de lavadoras u otros elementos. A lo largo de este proyecto he probado 3 tipos de bombas de agua diferentes.

La primera de ellas una bomba de agua de coche, la bomba de agua de los limpiaparabrisas, una bomba que funciona a 12v. Es una bomba que tiene un buen flujo de agua y tira el agua con bastante presión. Según la utilidad que le deseemos dar podría ser acertado su uso. Para el riego de esta huerta en concreto no la utilizaremos, pero en las pruebas que he realizado la he utilizado y en según qué riego se podría utilizar perfectamente.



Otra de las bombas que he usado es una de la marca Upright, concretamente la UP-7000. Esta bomba se puede utilizar para usos tan concretos como ósmosis inversa incluso, una bomba muy buena. En mi caso me la regalaron.

Esta bomba se alimenta con 24v, 36v o 48v. En mi caso utilizaré una fuente de alimentación de 24v.



Dimensiones		173x102x103 (mm)	
Peso		2 (kg)	
Voltaje	24 VDC	36 VDC	48 VDC
Rotación		550-600 rpm	
Temperatura de funcionamiento		10-40 C	
Test de aislamiento de voltaje		0.8Kv / 3 seg (sobre 0,6mA)	
Consumo	24 W	36 W	48 W
Flujo @ 80 psi	0.65 lpm	1.24 lpm	2.0 lpm

4.5. Electroválvula

En caso de que dispongamos de una toma de agua, podremos optar por colocar una electroválvula para el llenado del depósito que elijamos. O incluso teniendo una toma de red con suficiente presión para nuestro sistema de riego, podríamos utilizar solamente una electroválvula, con eso nos ahorraríamos el sensor de nivel, el depósito y la bomba del depósito de agua.

Yo hice pruebas con una electroválvula que aguenta 10 bares que muestro a continuación.



4.6. Depósitos de agua

Uno de los depósitos de agua que está previsto que se utilice es un bidón de 200 litros. El bidón tiene una altura de 875mm, un diámetro exterior de 585 y 18kg de peso.

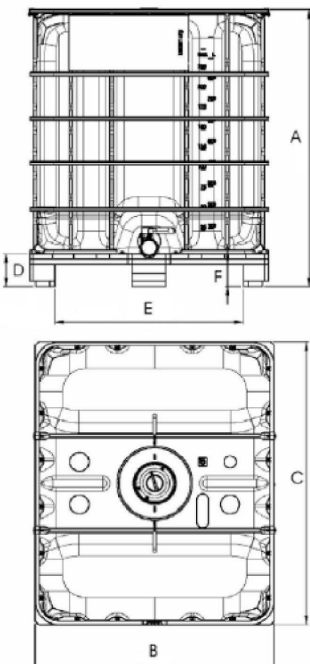


La tapa del bidón dispone de dos tapones roscado de 2" y 3/4" de acero en la tapa superior. Por uno de los tapones se introducirá el sensor de ultrasonidos para controlar la altura del agua y por el otro se introducirá la manguera de la bomba que rellenara el bidón y también la manguera para el riego.

Este bidón es una opción muy interesante por su precio económico, a partir de 11€ en internet de segunda mano.

Otra opción de mucha más capacidad son los bidones de 1000l de la marca Ibc. Con este tipo de bidón aseguramos mucho más acumulación de agua.



PSS No.:	W310100001	Fecha de revisión:	28-02-2009																																																																																																																																																												
Artículo:	MRI® SM6 STD, palet de madera tratado, boca de llenado DN150 (CCS160x7), válvula soldada de vaciado DN50, tapa: ciega con junta de EPDM, junta de la válvula: ETFE																																																																																																																																																														
Nº de plano:	M.4828.8																																																																																																																																																														
																																																																																																																																																															
<table> <tr> <th>Dimensiones</th><th>[mm]</th></tr> <tr> <td>altura 'A' ± 6</td><td>1160</td></tr> <tr> <td>ancho 'B' ± 6</td><td>1000</td></tr> <tr> <td>largo 'C' ± 6</td><td>1200</td></tr> <tr> <td>altura del palet 'D' ±</td><td>130</td></tr> <tr> <td>entrada palet 'E' ±</td><td>765</td></tr> <tr> <td>altura ent. Palet 'F' ±</td><td>72</td></tr> <tr> <td>Boca de llenado – CCS 160 x 7</td><td></td></tr> </table>	Dimensiones	[mm]	altura 'A' ± 6	1160	ancho 'B' ± 6	1000	largo 'C' ± 6	1200	altura del palet 'D' ±	130	entrada palet 'E' ±	765	altura ent. Palet 'F' ±	72	Boca de llenado – CCS 160 x 7																																																																																																																																																
Dimensiones	[mm]																																																																																																																																																														
altura 'A' ± 6	1160																																																																																																																																																														
ancho 'B' ± 6	1000																																																																																																																																																														
largo 'C' ± 6	1200																																																																																																																																																														
altura del palet 'D' ±	130																																																																																																																																																														
entrada palet 'E' ±	765																																																																																																																																																														
altura ent. Palet 'F' ±	72																																																																																																																																																														
Boca de llenado – CCS 160 x 7																																																																																																																																																															
<p>* pueden asimilarse otros líquidos</p> <p>** tener precaución acerca de la resistencia del material de la junta en función del líquido. otras juntas disponibles.</p> <p>*** puede cubrir otros líquidos, distintos a los ensayados</p> <p>**** existen materias primas alternativas, de acuerdo con la homologación</p>																																																																																																																																																															
<table> <tr> <th>IBC</th><th></th><th></th><th></th></tr> <tr> <td>Diseño:</td><td>Jaula metálica con cuerpo interior de PE-HD y palet de madera</td><td></td><td></td></tr> <tr> <td>capacidad / capacidad a rebose [l]</td><td>1000</td><td>1050</td><td></td></tr> <tr> <td>Tara [kg]</td><td>58</td><td></td><td></td></tr> <tr> <td>Altura máxima de apilado (a 25 °C) / a una densidad relativa [g/cm³]:</td><td>3 contenedores / 1.3</td><td>2 contenedores / 1.9</td><td></td></tr> <tr> <td>Certificación UN líquidos patrones: * **</td><td>Densidad relativa [g/cm³]</td><td></td><td></td></tr> <tr> <td>agua: * **</td><td>--</td><td>--</td><td>--</td></tr> <tr> <td>ácido acético: * **</td><td>--</td><td>--</td><td>--</td></tr> <tr> <td>solución tensoactiva: * **</td><td>--</td><td>--</td><td>--</td></tr> <tr> <td>acetato de n-butilo: * **</td><td>--</td><td>--</td><td>--</td></tr> <tr> <td>mezcla de hidrocarburos: * **</td><td>--</td><td>--</td><td>--</td></tr> <tr> <td>ácido nítrico 55%: * **</td><td>--</td><td>--</td><td>--</td></tr> <tr> <td>homologación: ***</td><td>--</td><td></td><td></td></tr> <tr> <th>Cuerpo interior</th><th></th><th></th><th></th></tr> <tr> <td>material</td><td>polietileno HD-PE estabilizado UV</td><td></td><td></td></tr> <tr> <td>tipo de material / color:</td><td>****</td><td>natural</td><td></td></tr> <tr> <td>fabricación:</td><td>extrusión soplado</td><td></td><td></td></tr> <tr> <td>peso cuerpo interior [kg]:</td><td>14</td><td></td><td></td></tr> <tr> <td>espesor mínimo [mm]:</td><td>1.2</td><td></td><td></td></tr> <tr> <td>tipo de cuello/paso [mm]:</td><td>CCS 160</td><td>7</td><td></td></tr> <tr> <td>diámetro interior de boca [mm]:</td><td>145</td><td></td><td></td></tr> <tr> <th>Tapa roscada</th><th></th><th></th><th></th></tr> <tr> <td></td><td>tapa</td><td>junta</td><td>tapón t-junta</td></tr> <tr> <td>material: **</td><td>PE-HD</td><td>EPDM</td><td>-- --</td></tr> <tr> <td>peso [g]:</td><td>210</td><td>35</td><td>-- --</td></tr> <tr> <td>sistema de desgasificación:</td><td>--</td><td></td><td></td></tr> <tr> <th>Válvula de vaciado</th><th></th><th></th><th></th></tr> <tr> <td>tipo:</td><td>válvula de mariposa soldada</td><td></td><td></td></tr> <tr> <td>material:</td><td>PE-HD</td><td></td><td></td></tr> <tr> <td>material de la junta: **</td><td>ETFE / PE</td><td></td><td></td></tr> <tr> <td>diámetro interior [mm]:</td><td>48</td><td></td><td></td></tr> <tr> <td>tipo rosca salida / paso:</td><td>--</td><td>--</td><td></td></tr> <tr> <th>Palet</th><th></th><th></th><th></th></tr> <tr> <td>material:</td><td>madera</td><td></td><td></td></tr> <tr> <td>fabricación:</td><td>clavado</td><td></td><td></td></tr> <tr> <td>tipo:</td><td>runner pallet</td><td></td><td></td></tr> <tr> <th>Jaula</th><th></th><th></th><th></th></tr> <tr> <td>material:</td><td>acero galvanizado</td><td></td><td></td></tr> <tr> <td>fabricación:</td><td>jaula de tubo soldada</td><td></td><td></td></tr> </table>	IBC				Diseño:	Jaula metálica con cuerpo interior de PE-HD y palet de madera			capacidad / capacidad a rebose [l]	1000	1050		Tara [kg]	58			Altura máxima de apilado (a 25 °C) / a una densidad relativa [g/cm³]:	3 contenedores / 1.3	2 contenedores / 1.9		Certificación UN líquidos patrones: * **	Densidad relativa [g/cm³]			agua: * **	--	--	--	ácido acético: * **	--	--	--	solución tensoactiva: * **	--	--	--	acetato de n-butilo: * **	--	--	--	mezcla de hidrocarburos: * **	--	--	--	ácido nítrico 55%: * **	--	--	--	homologación: ***	--			Cuerpo interior				material	polietileno HD-PE estabilizado UV			tipo de material / color:	****	natural		fabricación:	extrusión soplado			peso cuerpo interior [kg]:	14			espesor mínimo [mm]:	1.2			tipo de cuello/paso [mm]:	CCS 160	7		diámetro interior de boca [mm]:	145			Tapa roscada					tapa	junta	tapón t-junta	material: **	PE-HD	EPDM	-- --	peso [g]:	210	35	-- --	sistema de desgasificación:	--			Válvula de vaciado				tipo:	válvula de mariposa soldada			material:	PE-HD			material de la junta: **	ETFE / PE			diámetro interior [mm]:	48			tipo rosca salida / paso:	--	--		Palet				material:	madera			fabricación:	clavado			tipo:	runner pallet			Jaula				material:	acero galvanizado			fabricación:	jaula de tubo soldada					
IBC																																																																																																																																																															
Diseño:	Jaula metálica con cuerpo interior de PE-HD y palet de madera																																																																																																																																																														
capacidad / capacidad a rebose [l]	1000	1050																																																																																																																																																													
Tara [kg]	58																																																																																																																																																														
Altura máxima de apilado (a 25 °C) / a una densidad relativa [g/cm³]:	3 contenedores / 1.3	2 contenedores / 1.9																																																																																																																																																													
Certificación UN líquidos patrones: * **	Densidad relativa [g/cm³]																																																																																																																																																														
agua: * **	--	--	--																																																																																																																																																												
ácido acético: * **	--	--	--																																																																																																																																																												
solución tensoactiva: * **	--	--	--																																																																																																																																																												
acetato de n-butilo: * **	--	--	--																																																																																																																																																												
mezcla de hidrocarburos: * **	--	--	--																																																																																																																																																												
ácido nítrico 55%: * **	--	--	--																																																																																																																																																												
homologación: ***	--																																																																																																																																																														
Cuerpo interior																																																																																																																																																															
material	polietileno HD-PE estabilizado UV																																																																																																																																																														
tipo de material / color:	****	natural																																																																																																																																																													
fabricación:	extrusión soplado																																																																																																																																																														
peso cuerpo interior [kg]:	14																																																																																																																																																														
espesor mínimo [mm]:	1.2																																																																																																																																																														
tipo de cuello/paso [mm]:	CCS 160	7																																																																																																																																																													
diámetro interior de boca [mm]:	145																																																																																																																																																														
Tapa roscada																																																																																																																																																															
	tapa	junta	tapón t-junta																																																																																																																																																												
material: **	PE-HD	EPDM	-- --																																																																																																																																																												
peso [g]:	210	35	-- --																																																																																																																																																												
sistema de desgasificación:	--																																																																																																																																																														
Válvula de vaciado																																																																																																																																																															
tipo:	válvula de mariposa soldada																																																																																																																																																														
material:	PE-HD																																																																																																																																																														
material de la junta: **	ETFE / PE																																																																																																																																																														
diámetro interior [mm]:	48																																																																																																																																																														
tipo rosca salida / paso:	--	--																																																																																																																																																													
Palet																																																																																																																																																															
material:	madera																																																																																																																																																														
fabricación:	clavado																																																																																																																																																														
tipo:	runner pallet																																																																																																																																																														
Jaula																																																																																																																																																															
material:	acero galvanizado																																																																																																																																																														
fabricación:	jaula de tubo soldada																																																																																																																																																														
Otros componentes:	salida válvula con disco de aluminio, vertedero, etiqueta recogida, 1 placa delantera 235x500																																																																																																																																																														

Para este tipo de depósitos tenemos multitud de accesorios que se pueden adquirir. En caso de necesitar dos salidas en vez de una podemos comprar este tipo de accesorios para poner dos circuitos independientes.



Si solo necesitamos una sola salida para una bomba utilizaríamos el siguiente accesorio



4.7. Led SMD de 3 colores

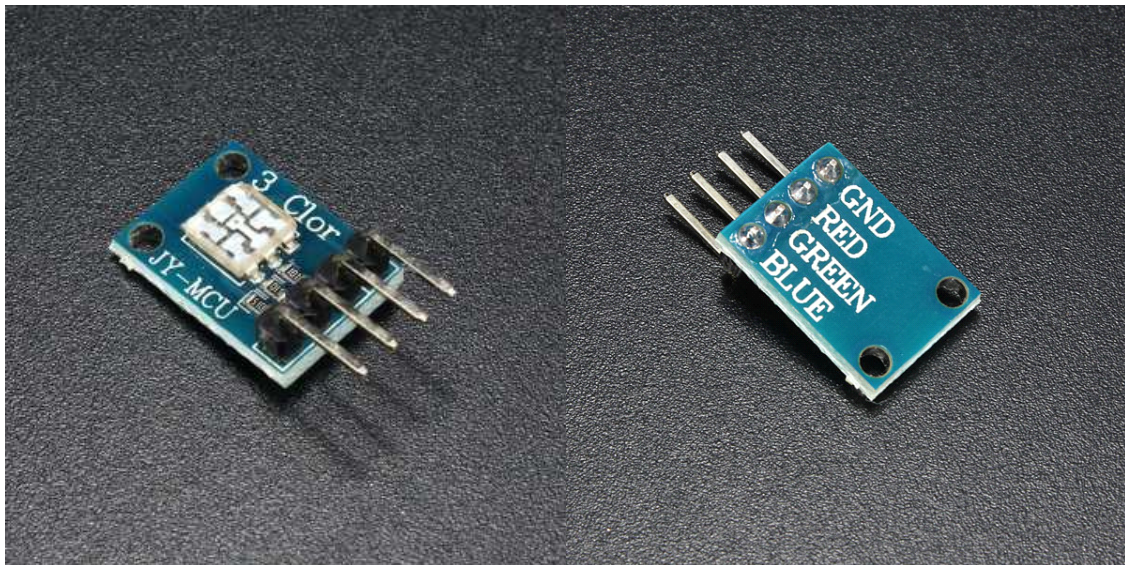
Una pega que se le podría encontrar a este sistema de riego, es que no muestra nada de información para que la persona que lo utilice sepa si está funcionando correctamente el riego o no. Las bombas se puede saber si están funcionando correctamente, puesto que podríamos escucharlas cuando estén funcionando, pero con los sensores y Arduino no podríamos estar seguros del todo de que funcionan correctamente.

Se podría colocar una pantalla que nos informe en todo momento de cómo evoluciona el programa y nos muestre las lecturas y demás acciones del programa. Pero como uno de los objetivos de este proyecto es que sea lo más barato posible y no invertir mucho dinero, la pantalla la descarte por su coste (más de 16€).

Al descartar la pantalla, se me ocurrió poner un led de 3 colores SMD, concretamente el RGB SMD 5050. Se puede comprar ya montado en una placa preparada para Arduino con las resistencias necesarias para proteger el led. Se pueden comprar por 2.5€ en internet.

Las conexiones se realizan mediante una patilla que es la GND y las otras tres patillas alimentan cada uno de los colores por independiente.

El programa encenderá el led en Rojo cuando la humedad es baja, en Verde cuando es correcta y en azul cuando la tierra tiene mucha humedad.



4.8. Caja estanca

Para proteger el sistema de riego, utilizaremos una caja estanca de la marca Legrand para introducir la placa Arduino y los relés que controlaran las bombas y el sensor de humedad. Introduciendo el Arduino y relés dentro nos aseguraremos de proteger el sistema de la humedad y posible contacto con líquidos. Por los orificios que tiene la caja estanca introduciremos la alimentación para la placa Arduino y las conexiones para las bombas y sacaremos el cableado del sensor de humedad.

La caja que utilizaré para el montaje en la huerta será esta



5. RIEGO POR GOTEO

5.1. ¿Qué es el riego por goteo?

El riego por goteo, es un método de irrigación utilizado en las zonas áridas pues permite la utilización óptima de agua y abonos.

Consiste en la infiltración lentamente de agua hacia las raíces de las plantas e irrigando directamente la zona de influencia de las raíces, a través de un sistema de tuberías y emisores (goteros a baja presión) consiguiendo así reducir al mínimo la utilización de agua.

Esta técnica es la innovación más importante en agricultura. Ahorrando en un promedio entre el 40 y el 60% de agua con respecto a los sistemas de riego tradicionales.

5.2. Historia

El riego por goteo se ha utilizado desde la antigüedad cuando se enterraban vasijas de arcilla llenas de agua con el fin de que el agua se infiltrara gradualmente en el suelo. El riego gota a gota moderno se desarrolló en Israel porque el país tenía escasas de agua, querían aprovechar cada gota. En los años 40 el célebre granjero mexicano Máximo Alonzo perfeccionó el sistema, llevándolo a su máxima capacidad de expresión, tal como lo conocemos hoy día.

Con la llegada de los plásticos modernos después de la Segunda Guerra Mundial, fueron posibles numerosas mejoras. Micro-tubos de plástico y diversos tipos de goteros han sido empleados en invernadero en Europa y en Estados Unidos.

La moderna tecnología de riego por goteo fue inventada en Israel por Simcha Blass y su hijo Yeshayahu. En lugar de liberar el agua por agujeros minúsculos, que fácilmente se podían obstruir por acumulación de partículas minúsculas, el agua se libera por tuberías más grandes y más largas empleando el frotamiento para ralentizar la velocidad del agua en el interior de un emisor (gotero) de plástico. El primer sistema experimental de este tipo fue establecido en 1959 cuando la familia de Blass en el Kibboutz Hatzerim creó una compañía de riegos llamada Netafim.

A continuación, desarrollaron y patentaron el primer emisor exterior de riego por gota a gota. En 1976, Gershon Eckstein (empresa DIS) inventa la máquina extrusora de goteros, eliminando la necesidad de insertar los goteros en el campo. Posteriormente, los enrolladores automáticos permitieron acelerar la velocidad de fabricación por encima de los 65m/min. En los años 90, el desarrollo del gotero antidrenante y antisucción, permitiendo el desarrollo del riego subterráneo.

En la actualidad, se ha concentrado la fabricación de tuberías emisoras en un pequeño número de fabricantes: Netafim; NaandanJain (Chapin, Thomas Machine, Point Source Irrigation); John Deere (Roberts, T-System, Plastro), AZUD [\[1\]](#) y otras como Irritec o Eurodrip.

Por otra parte, la fabricación del gotero autocompensante PC se limita a EEUU, Israel, Italia y España.

5.3. Evolución del riego por goteo

En la evolución del riego por goteo se espera el desarrollo de la fertirrigación paralelo al riego por goteo (existe una amplia gama de fertilizantes que encuentra en este sistema la vía más eficiente para su aplicación).

A lo largo del tiempo se han ido añadiendo numerosas mejoras para solucionar y evitar problemas que podría tener el sistema:

- **Goteros autocompensantes:** dan un caudal más o menos fijo dentro de unos márgenes de presión. Es útil para que los goteros del final del tubo no den menos agua que los del principio debido a la caída de presión debida al rozamiento. También son útiles cuando el tubo va en cuesta. Los goteros más bajos soportaran más presión y si no son adecuados pueden perder demasiada agua.
- **Goteros y filtros autolimpiables:** este sistema de riego es muy sensible a las partículas sólidas y se suelen instalar filtros muy eficaces y con sistemas de autolimpieza periódica. Los propios goteros también pueden tener un sistema para eliminar pequeñas partículas que puedan atascarlos.
- **Goteros regulables:** se puede regular el caudal con un mando mecánico.

La mayor parte de los grandes sistemas de riego por goteo utilizan un cierto tipo de filtro de agua para impedir la obstrucción de los pequeños tubos surtidores. Ciertos sistemas utilizados en zonas residenciales se instalan sin filtros adicionales ya que el agua potable ya está filtrada. Prácticamente todos los fabricantes de equipos de riego por goteo recomiendan que se utilicen los filtros y generalmente no dan garantías a menos que esto sea hecho.

El riego por goteo se emplea casi exclusivamente utilizando agua potable pues las reglamentaciones desaconsejan generalmente pulverizar agua no potable. En riego por goteo, la utilización de abonos tradicionales en superficie es casi ineficaz, así los sistemas de goteo mezclan a menudo el abono líquido o pesticidas en el agua de riego. Otros productos químicos tales como el cloro o el ácido sulfúrico son igualmente utilizados para limpiar periódicamente el sistema.

Si está correctamente montado, instalado, y controlado, el riego por goteo puede ayudar a realizar importantes ahorros de agua por la reducción de la evaporación. Por otro lado, el riego por goteo puede eliminar muchas enfermedades que nacen del contacto del agua con las hojas. En conclusión, en las regiones donde los aprovisionamientos de agua están muy limitados, se puede obtener un notable aumento de producción utilizando la misma cantidad de agua que antes.



5.4. Ventajas del riego por goteo

El riego por goteo es un medio eficaz y pertinente de aportar agua a la planta, ya sea en cultivos en línea (mayoría de los cultivos hortícolas o bajo invernadero, viñedos) o en plantas (árboles) aisladas (vergeles). Este sistema de riego presenta diversas ventajas desde los puntos de vista agronómicos, técnicos y económicos, derivados de un uso más eficiente del agua y de la mano de obra. Además, permite utilizar caudales pequeños de agua.

- Una importante reducción de la evaporación del suelo, lo que trae una reducción significativa de las necesidades de agua al hacer un uso más eficiente gracias a la localización de las pequeñas salidas de agua, donde las plantas más las necesitan. No se puede hablar de una reducción en lo que se refiere a la transpiración del cultivo, ya que la cantidad de agua transpirada (eficiencia de transpiración) es una característica fisiológica de la especie.
- La posibilidad de automatizar completamente el sistema de riego, con los consiguientes ahorros en mano de obra. El control de las dosis de aplicación es más fácil y completo. Se pueden utilizar aguas más salinas que en riego convencional, debido al mantenimiento de una humedad relativamente alta en la zona radical (bulbo húmedo).
- Una adaptación más fácil en terrenos rocosos o con fuertes pendientes.
- Reduce la proliferación de malas hierbas en las zonas no regadas
- Permite el aporte controlado de nutrientes con el agua de riego sin pérdidas por lixiviación con posibilidad de modificarlos en cualquier momento del cultivo. (fertirriego)
- Permite el uso de aguas residuales ya que evita que se dispersen gotas con posibles patógenos en el aire.
- Incremento notable en la producción. Adaptación a todo tipo de superficies y desniveles en su relieve natural sin inversión en la nivelación y transporte de tierras. Reduce la proliferación de malas hierbas en las zonas no regadas. Permite el aporte controlado de nutrientes con el agua de riego.

5.5. Inconvenientes del riego por goteo

Sus principales inconvenientes son:

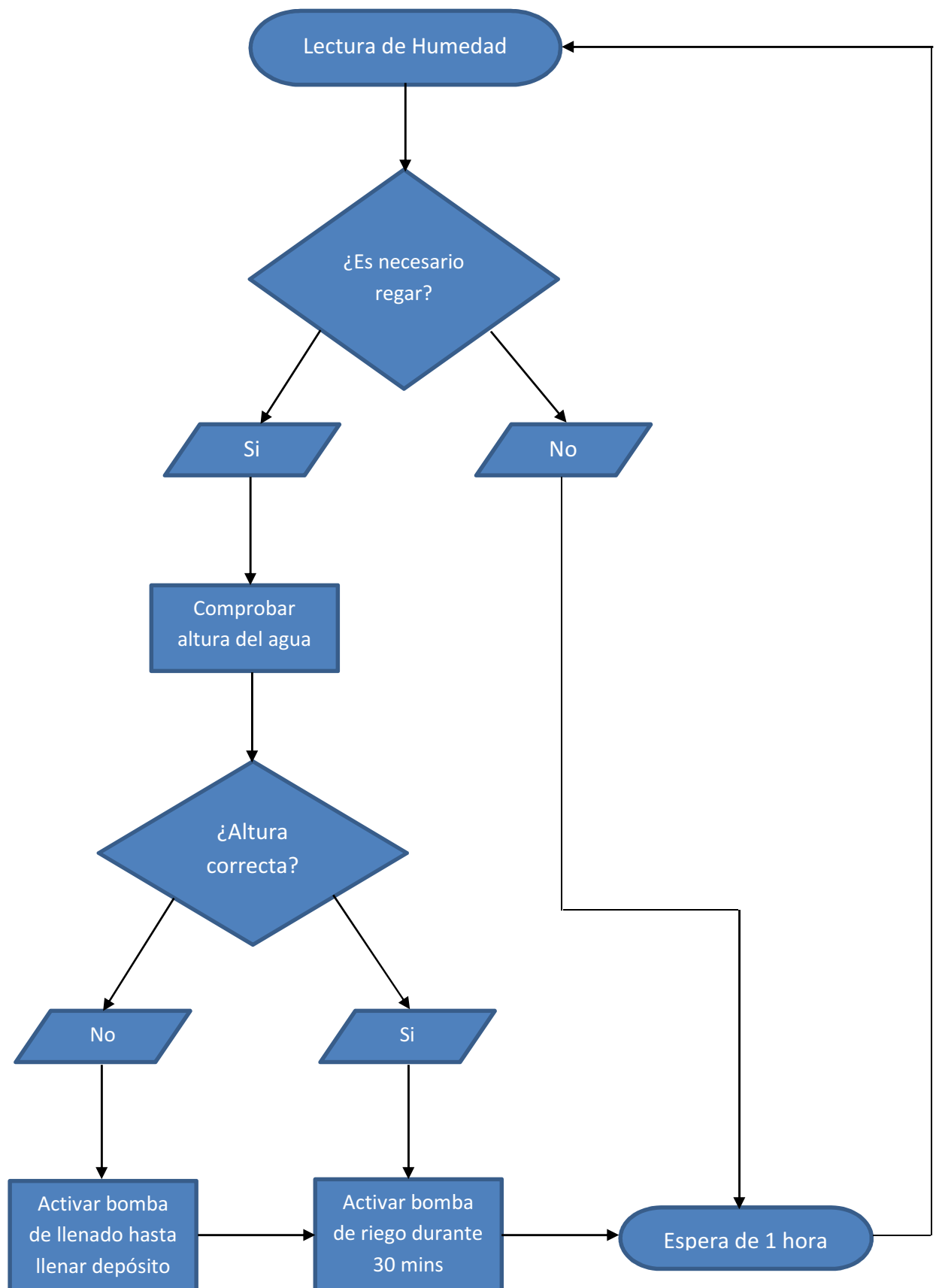
- El coste elevado de la instalación. Se necesita una inversión elevada debida a la cantidad importante de emisores, tuberías, equipamientos especiales en el cabezal de riego y la casi necesidad de un sistema de control automatizado (electro-válvulas, programador). Sin embargo, el aumento relativo de coste con respecto a un sistema convencional no es prohibitivo.
- El alto riesgo de obturación ("clogging" en inglés) de los emisores, y el consiguiente efecto sobre la uniformidad del riego. Esto puede ser considerado como el principal problema en riego por goteo. Por ello en este sistema de riego es muy importante el sistema de filtración implantado, que dependerá de las características del agua utilizada. De hecho hay sistemas que funcionan con aguas residuales y aguas grises.
- La presencia de altas concentraciones de sales alrededor de las zonas regadas, debida a la acumulación preferencial en estas zonas de las sales. Esto puede constituir un inconveniente importante para la plantación siguiente, si las lluvias no son suficientes para lavar el suelo.
- Un inconveniente muy importante de este sistema tan particular, es el tapado de los orificios, por lo tanto no regarán como nosotros esperamos.

5.6. Tabla de riego de hortalizas

	Acelgas	Lechugas	Tomates	Patatas	Espinacas	Berenjena	Brócoli	Zanahoria
Temporada de cultivo	Primavera, otoño e invierno	Anual	Finales de la primavera	Primavera	Todo el año	Invierno	Primavera (abril – junio), verano (agosto - septiembre)	Primavera, otoño e invierno
Profundidad raíces	25cm	10cm	40cm	7-8cm	2cm	25-30cm		25cm
Humedad	60-70% de humedad, no admite exceso	2 veces por semana, 60-80% de humedad	65-70% de humedad	85-90% de humedad	90% humedad	90% humedad	60-75%	Porte diario de corta duración
Riego	2 o 3 veces a la semana, esto dependiendo de qué tan caluroso esté el clima. Lo ideal es darle buena cantidad de agua en cada riego.	No regar abundante para evitar hongos	Gran cantidad de riego, no muy frecuente	Muy exigente en agua. preferible por la noche	Cortos y frecuentes, especialmente en las últimas fases del cultivo	Abundante y regular	Humedad prácticamente constante, aportes de agua abundantes y regulares	Mantener la tierra siempre húmeda
Como cultivar	El huerto, 8 cm de distancia entre planta y planta y 45cm entre líneas	30 cm entre líneas y 20-30 entre planta	60cn de distancia entre planta y otra 90cn entre líneas	Distancia entre una planta y otra 30cm Distancia entre líneas 75cm	10cm entre planta y planta, 30cm entre líneas	Distancia entre plantas 60cm, 80cm entre líneas		Distancia entre plantas 8 cm entre líneas 30cm
Observaciones		Tolera mejora las temperaturas bajas que altas. - 6 a 30°C		Se debe evitar sembrar en zonas con viento	Siembra de la semilla directo en tierra a 2 cm de profundidad			Regar antes de plantas. Temperatura 16-18°C

	Judías Verdes	Cebolla	Puerro	Pepino	Ajo	Alcachofa	Guisantes	Pimientos
Temporada de cultivo	Otoño y primavera	Primavera	Primavera	Primavera	Primavera y otoño	Verano	Invierno-primavera	Primavera
Profundidad raíces	8cm	15-20cms	20cms	30cm	5cm		5cm	
Humedad	80%	>60%	80%	80%	70-80%		60%	50-70%
Riego	Riego frecuente	Riego regular, sobre todo durante la formación del bulbo	Riego constante	Riego abundante	Siempre húmedo pero sin encharcamiento	Riego frecuente y no demasiado abundante		Moderado y constante
Como cultivar	15cm entre planta y 60cm entre líneas	10cm entre plantas y 20cm entre líneas	8cm entre planta y 30cm entre líneas	12cm entre plantas y 15cm entre líneas	15cm entre plantas y 30cm entre líneas			40-50cm entre plantas
Observaciones		Se adapta a todas las temperaturas		Tierra húmeda pero no encharcada				

6. DIAGRAMA DE FLUJO



7. MONTAJE Y PUESTA EN MARCHA

El sistema de riego estará dentro de una caseta que se situará dentro de la huerta. Esta caseta aún no está construida, y tan pronto como se construya se realizará la instalación de todo el sistema de riego en su interior para evitar que todo el equipo sea robado.

Dentro de la caseta el Arduino se instalará dentro de una caja estanca Legrand, colocando dentro de la misma tanto el Arduino como los relés.

El depósito de agua que se utilizará se decidirá en el momento en el que se construya la caseta y se decida concretamente el terreno que se utilizará.

Para la alimentación del sistema de riego, en la huerta dispondremos de una toma de 230v de corriente alterna en todo momento, la cual utilizaremos para conectar la fuente de alimentación de 24V de corriente continua para alimentar la bomba, la fuente de 9V del Arduino y la bomba de 230v de llenado del depósito.

7.1. Montaje del sensor de humedad

El sensor de humedad lo tendremos que poner cerca de la zona donde el sistema de riego por goteo actúa, es decir cerca de lo que hayamos sembrado y de la gota que cae del sistema de riego. A su vez, según las raíces de lo que tengamos sembrado tendremos que enterrar más o menos el sensor en el terreno para que realice correctamente las mediciones. El sensor lo colocaremos de manera vertical.

7.2. Ajustes el programa para Arduino

A la hora de diseñar el programa, he añadido una serie de variables, que se declaran al principio del programa y nos permiten ajustar tanto las alturas máximas y mínimas del tanque de agua como la humedad mínima, a partir de la cual el riego comenzara a funcionar, y también la humedad máxima, a partir de la cual el sistema nos informará de que el suelo tiene mucha humedad.

Las primeras cuatro líneas del programa serán estas variables que deberemos de modificar a nuestro caso:

```
int distancia_hasta_max= 0;    //Distancia desde el sensor hasta el nivel max en CM
int distancia_hasta_min= 0;    //Distancia desde el sensor hasta el nivel min en CM
int humedad_min = 300;        //Humedad mínima para que empiece a regar
int humedad_max = 700;        //Humedad a partir de la cual consideramos mucha
long tiempo_riego = 1800000;  // Tiempo en milisegundos de riego
long tiempo_espera = 3600000; // Tiempo de espera en milisegundos
```

Cambiando los valores después del “=” por el valor que nosotros deseemos, el resto del código del programa funcionará sin ningún problema.

distancia_hasta_min, será la distancia desde el sensor hasta la altura mínima del tanque en cm. Con esta altura se podrá efectuar un riego de 30 minutos con la bomba que tengamos instalada. Esto deberemos de calcularlo, teniendo en cuenta el caudal de la bomba y el depósito que elijamos. La altura podremos configurarla según nuestras necesidades, pero se debe calcular correctamente para evitar que la bomba funcione en vacío.

distancia_hasta_max, será la distancia desde el sensor hasta la altura máxima del agua en el tanque. Dejar una distancia mínima de unos 10cm para que el agua no entre en contacto con el sensor de ultrasonidos situado en el interior del tanque.

La variable *humedad_min*, será la humedad mínima que tendrá nuestro huerto en todo momento. Por lo general dejándolo en 300 es suficiente, pero si se quiere tener más humedad se aumentaría ese número, y si se quiere tener menos humedad se pone un número menor.

Humedad_max, es la humedad a partir de la cual el sistema nos informará de que el huerto tiene demasiada humedad y el led se encenderá en azul.

Tiempo_riego controla el tiempo que se regará, el tiempo tenemos que configurarlo en milisegundos, así como el *tiempo_espera*, que es el tiempo que transcurre entre mediciones de humedad. Por defecto el tiempo de riego está configurado en 30 minutos y el tiempo de espera entre mediciones 60 minutos.

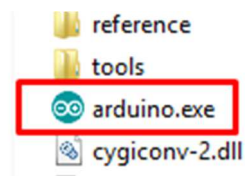
7.3. Puesta en marcha

Una vez tengamos todo el montaje siguiendo detalladamente los planos, y los valores descritos en el apartado anterior para el programa, lo siguiente que tendremos que hacer es programar el Arduino Leonardo (o cualquier otro Arduino que disponga de las conexiones necesarias para el montaje que vamos a realizar).

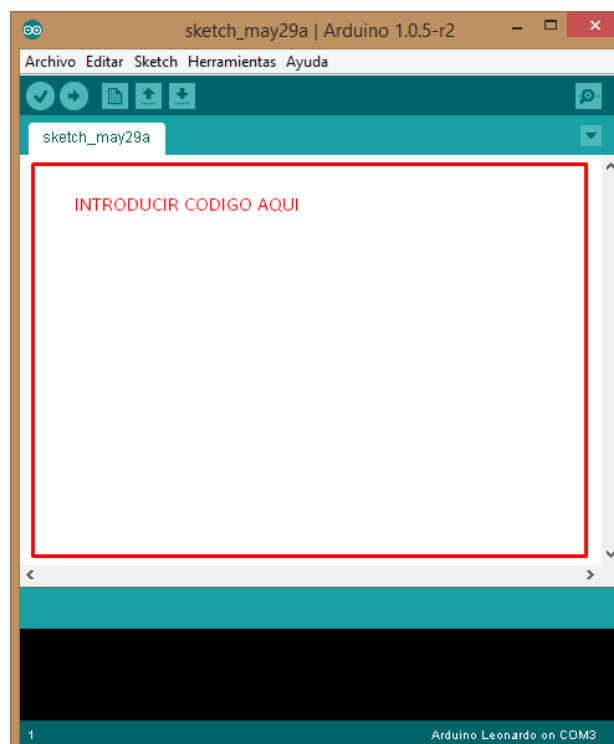
Lo primero que tenemos que hacer es descargarnos el programa de la página de Arduino según el sistema que utilicemos de la siguiente dirección: <http://arduino.cc/en/Main/Software>

Aquí utilizaremos la versión para Windows que viene en un archivo Zip, una vez descargamos el archivo, descomprimos el contenido del Zip, por ejemplo, en el escritorio. Para meter el programa en el Arduino tenemos que enchufar el Arduino con un cable micro-USB al puerto USB de ordenador y seguir estos pasos:

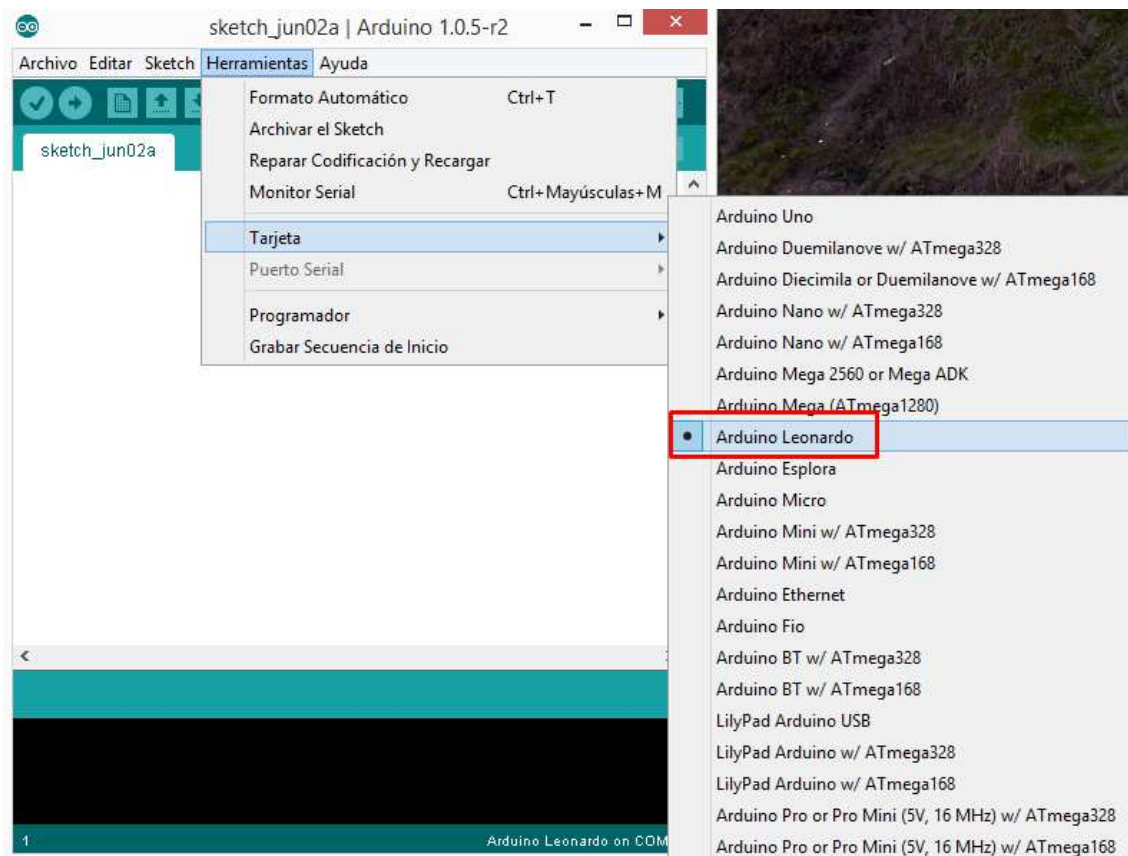
- 1- Lo primero, si no tenemos los drivers del Arduino instalados, al conectarlo al USB del ordenador instalaremos los drivers, situados en la carpeta DRIVERS dentro de nuestra carpeta de Arduino.
- 2- Una vez tenemos los drivers instalados ejecutamos el programa



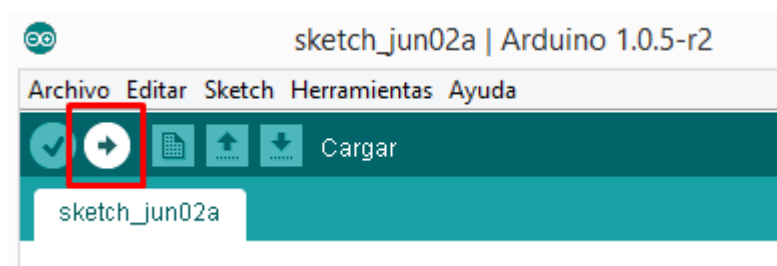
- 3- Cuando estamos en la pantalla principal del programa tenemos que escribir el código en la pantalla (El código está en el punto “10. Código Arduino”).



- 4- Con el código ya finalizado tenemos que ir a Herramientas → Tarjeta y seleccionamos nuestro Arduino, en este caso ARDUINO LEONARDO



- 5- Para finalizar la configuración del programa, en Herramientas → puerto Serial, seleccionamos el puerto en el que nos ha instalado el Windows el Arduino, en mi caso el COM 3
- 6- Para escribir el programa en el Arduino, hacemos clic en el botón de CARGAR del programa y empezará a escribir el programa.



7.4. Montaje Huerta

Para el riego utilizaré una bomba, la Upright 7000. Es una bomba que funciona a 24V. Con esta bomba proporcionaré el caudal y la presión necesaria para que la tubería de riego por goteo funcione correctamente. En este caso se utilizarán 30m de tubería de riego de 16mm.

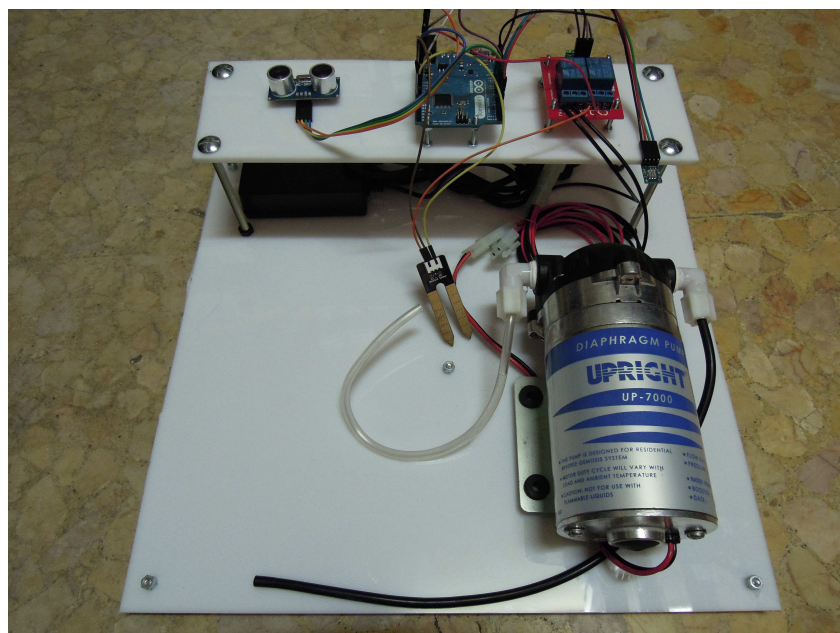
Podríamos poner otro tipo de motores, como bombas de lavadora, bombas de agua convencionales o cualquier otro que se nos ocurra, siempre que cumpla las especificaciones que necesitamos.

En cuanto al llenado del depósito de agua nos encontramos en la misma situación, si disponemos de un sitio donde coger el agua y tenemos que utilizar una bomba, podemos coger cualquier tipo de bomba de agua que queramos, ya que la podremos conectar sin ningún problema. El encargado de controlar el nivel del tanque del agua será el sensor de ultrasonidos, él será el encargado de medir el nivel del tanque y mediante la calibración del mismo y su posterior programación, ajustaremos en qué momento se tiene que activar/desactivar la bomba para el llenado del tanque de agua.

7.5. Maqueta

La maqueta que he realizado, en principio la hice para demostrar y comprobar que el sistema de riego funcionaba correctamente y poder observar los problemas o inconvenientes que pudiesen surgir.

He tenido la maqueta montada durante 3 meses continuamente funcionando con una planta para controlar su riego de manera autónoma. Como solo se trataba de una planta el sistema se simplificó un poco y solo utilice una bomba de riego y el sensor de humedad de tierra. El sensor de ultrasonidos también lo coloqué, pero solo para comprobar su correcto funcionamiento y no como sensor de nivel del tanque de riego.



8. SINOPSIS DEL PROYECTO

8.1. Propósito del proyecto

- 1- Crear un riego autónomo
- 2- Gastar el menor dinero posible sin comprometer el resultado
- 3- “Universalizarlo”: Que sistemas de riego ya implantados puedan ser adaptados a este automatismo
- 4- Posibilidad de reutilizar materiales al no estar cerrado a unos en concreto

8.2. Elección de materiales

8.2.6. Arduino

El Arduino que utilice es el Leonardo porque era el que más fácil podía conseguir y no era de los más caros. Se pueden comprar de marcas chinas muy baratos.

8.2.7. Sensores

Los sensores capacitivos los descarte por su alto precio, diez veces más caros que los resistivos.

Me decante por los sensores resistivos por su exactitud en las mediciones, su precio bajo y porque disponía de una versión que trae una placa con un comparador digital que me permite hacer tanto lecturas analógicas como digitales

También existe la posibilidad de crearnos nuestros propios sensores caseros. Disponemos en internet de esquemas que nos enseñan cómo realizarlos. Decidí decantarme por sensores comerciales, porque para gente poco experta en manualidades sería complicado.

El mantenimiento de estos sensores se debería de realizar cada 3-4 meses para asegurarnos el correcto funcionamiento y que las mediciones sean correctas.

8.2.8. Bombas de agua

Las bombas de agua se pueden utilizar las que se quieran.

- El llenado del depósito se monitoriza con el sensor de ultrasonidos en todo momento. Si se elige una bomba con menor caudal que otra, tardará más en llenar, eso es todo.
- La bomba de riego tiene que ser una que cumpla con las especificaciones del riego que se instale.

8.2.9. Electroválvula

Si se utiliza una toma de agua con presión para el llenado del depósito o el sistema de riego, tiene que aguantar la presión a la que se le someta.

8.2.10. Relé

Los relés que compre son unos que vienen preparados con las conexiones para Arduino. Baratos y de buena calidad.

8.2.11. Sensores de nivel

Para controlar el nivel del agua dentro del tanque de riego, primero pensé en sensores de flotador. Pero al final buscando sensores encontré el de ultrasonidos y me decante por este por varias razones:

- 1- Por su bajo precio
- 2- Fácil instalación (en la parte de arriba del depósito)
- 3- Evitaba agujerear cada vez que quería establecer nuevos niveles
- 4- Me facilitaba mucho realizar el programa de riego

8.2.12. Cableado

Para las conexiones con los sensores y relés utilicé cable que venden específico para Arduino y que es muy barato. Para las pruebas de colocar el sensor a 8 metros utilicé cable de par trenzado utp-5.

8.3. Montaje

Para el montaje del Arduino con los relés se utilizará una caja estanca, para proteger el Arduino y los relés de humedad y salpicaduras.

En el caso de las bombas de riego, se debe de hacer especial hincapié en la protección de los enchufes y conexiones contra la humedad y el agua.

La instalación del sensor de ultrasonidos se debe realizar en la parte más alta del depósito y centrado, “mirando” perfectamente al agua. Yo recomendaría utilizar silicona o pegamento termofusible para fijarlo y rodear todo el sensor (menos los elementos que utiliza para medir). Así nos aseguramos de proteger la circuitería del sensor de la humedad.

8.4. Otros aspectos a tener en cuenta

En este proyecto he utilizado un solo sensor de humedad. El sensor es muy preciso siempre que se coloque en el sitio correcto.

También se podrían colocar dos o tres sensores y realizar la media de las mediciones para tener una exactitud mayor de la humedad de la tierra. O poner sensores en diferentes tipos de cultivos para controlar con diferentes bombas diferentes trozos de terreno.

8.5. Código de programación

A la hora de realizar el programa intente crear una serie de variables para que cualquiera que no esté familiarizado con la programación pueda ajustar el programa a sus necesidades.

Una vez tenemos instalado el programa y realizadas todas las conexiones el funcionamiento es el siguiente:

- 1- Realiza una lectura de humedad y ajusta el color de led (Rojo-Seco, Verde-Húmedo, Azul- Muy Húmedo)
- 2- Si la humedad es superior a la mínima, pasa al punto 3. Si la humedad es inferior a la mínima:
 - a. Realiza otra medición para comprobar si la humedad es inferior a la mínima (y ajusta el led si es necesario). Si es superior a la mínima pasa al punto 3. Sino pasa al punto b
 - b. Realiza una comprobación de la altura del depósito. Si la altura está por debajo de la mínima, rellena el depósito hasta el máximo. Una vez termine esta comprobación efectúa el riego.
- 3- Espera un tiempo determinado por el usuario y vuelve al punto 1.

Aclaración: Una vez se realiza el riego, se espera un tiempo hasta hacer la comprobación de humedad para dejar que el agua penetre.

9. PRESUPUESTO

A continuación realizaré un presupuesto del gasto que supondría el sistema Arduino que propongo.

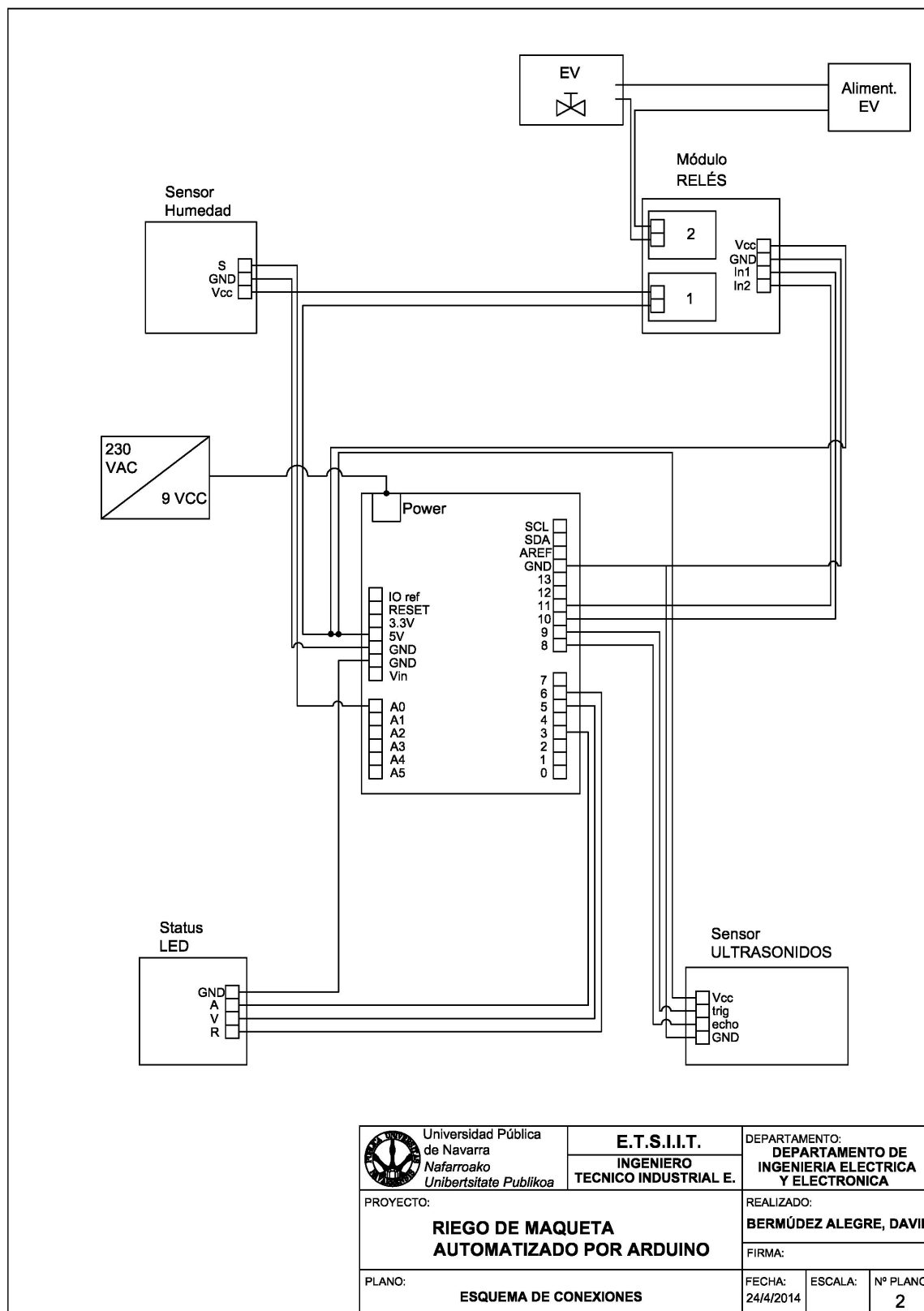
Material	Unidades	Precio
Arduino Leonardo	1	25€
Sensor humedad	1	3€
Led RGB	1	2,5€
Sensor ultrasonidos	1	2,5€
Cableado	1	12€
Placas relés	1	4€
Total		49€

Como podemos observar realizar el control de riego puede resultar muy barato y mucho más si lo comparamos con sistemas de riego de marcas conocidas que solo el programador HORARIO nos costaría más de 120€. Si le añadiríamos un sensor para detener el riego horario cuando llueve, nos supondría otros mas 60€ aproximadamente.

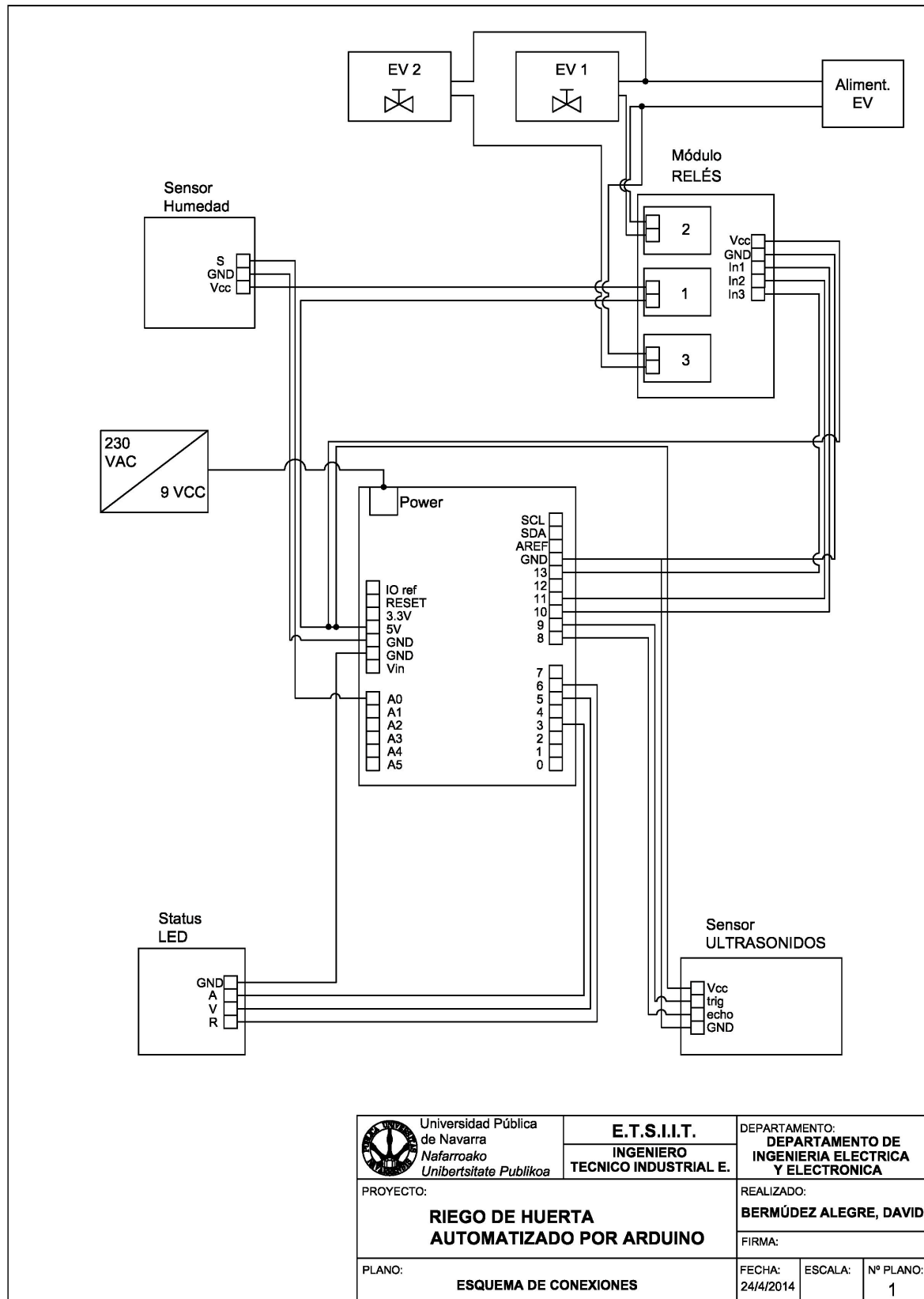


10. PLANOS

10.1. Plano maqueta



10.2. Plano Huerta



11. Código Arduino

11.1. Código de programación para la maqueta

```
int humedad_min = 300;      // Humedad minima para que empiece a regar
int humedad_max = 700;      // Humedad apartir de la cual se considera mucha humedad
long tiempo_riego = 5000;   // Tiempo en milisegundos de riego
long tiempo_espera = 15000; // Tiempo de espera en milisegundos

int redPin = 3;    // RED pin del LED a PWM pin 6
int greenPin = 5;  // GREEN pin del LED a PWM pin 5
int bluePin = 6;   // BLUE pin del LED a PWM pin 3
int echo = 8;      // Pin para recibir el pulso de eco
int trig = 9;      // Pin para enviar el pulso de disparo
int rele1 = 10;    // Rele en el pin 10
int rele2 = 11;    // Rele en el pin 11
int humedad = 0;
int rojo = 0;
int verde = 0;
int azul = 0;
int nivel = 0;

void setup()
{
  Serial.begin(57600);
  pinMode(echo, INPUT);
  pinMode(trig, OUTPUT);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(rele1, OUTPUT);
  pinMode(rele2, OUTPUT);
  digitalWrite(rele1, HIGH);
  digitalWrite(rele2, LOW);
}

void loop()
{
  lectura_humedad ();
  Sensor_Distancia ();
  if (humedad < humedad_min)
  {
    Riego (humedad);
  }
  delay(tiempo_espera);
}

void lectura_humedad (void)
{
  digitalWrite(rele1, HIGH);
  delay(2000);
  Serial.print("Humedad de la tierra:");
  humedad = analogRead (0);
  humedad = 1023 - humedad;
  Serial.println(humedad);
  digitalWrite(rele1, LOW);

  Color_Led (humedad);
}

void Color_Led (int humedad )
```

```

{
if (humedad >= 0 and humedad < humedad_min)
{
    analogWrite(redPin, 255);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);
}
if (humedad >= humedad_min and humedad <= humedad_max)
{
    analogWrite(redPin, 0);
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 0);
}
if (humedad > humedad_max)
{
    analogWrite(redPin, 0);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 255);
}
}

void Sensor_Distancia(void)
{
    unsigned int tiempo, distancia;
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);          // Calcula la distancia midiendo el tiempo del estado alto
    del pin ECHO
    tiempo = pulseIn(echo, HIGH);     // La velocidad del sonido es de 340m/s o 29
    microsegundos por centimetro
    distancia= tiempo/58;             //manda la distancia al monitor serie
    Serial.print(distancia);
    Serial.println(" cm");
}

void Riego (int )
{
    lectura_humedad ();              // Realializo una lectura para confirmar que la
    medición es correcta y realmente es bajo
    if (humedad < humedad_min)
    {
        digitalWrite(rele1, LOW);
        digitalWrite (rele2, HIGH);
        Serial.println("Regando");
        delay (tiempo_riego);
        digitalWrite (rele2, LOW);
    }
    else
    {
        Serial.println("No regar");
    }
}
}

```


11.2. Código de programación para la huerta

```
int distancia_hasta_max= 0;    //Distancia desde el sensor hasta el nivel max en CM
int distancia_hasta_min = 0;    //Distancia desde el sensor hasta el nivel min en CM
int humedad_min = 300;         //Humedad mínima para que empiece a regar
int humedad_max = 700;         //Humedad a partir de la cual se considera mucha humedad
long tiempo_riego = 1800000;   //Tiempo en milisegundos de riego
long tiempo_espera = 3600000;  // Tiempo de espera en milisegundos

int redPin = 3;    // RED pin del LED a PWM pin 6
int greenPin = 5;  // GREEN pin del LED a PWM pin 5
int bluePin = 6;   // BLUE pin del LED a PWM pin 3
int echo = 8;      // Pin para recibir el pulso de eco
int trig = 9;      // Pin para enviar el pulso de disparo
int rele1 = 10;    // Rele en el pin 10 del sensor de humedad
int rele2 = 11;    // Rele en el pin 11 de la bomba de agua
int rele3 = 12;    // Rele en el pin 12 de la E.V.
int distancia = 0;
int humedad = 0;
int rojo = 0;
int verde = 0;
int azul = 0;
int nivel = 0;

void setup()
{
  Serial.begin(57600);
  pinMode(echo, INPUT);
  pinMode(trig, OUTPUT);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(rele1, OUTPUT);
  pinMode(rele2, OUTPUT);
  digitalWrite(rele1, HIGH);
  digitalWrite(rele2, LOW);
  digitalWrite(rele3, LOW);
}

void loop()
{
  lectura_humedad ();
  Sensor_Distancia ();
  if (humedad < humedad_min)
  {
    Riego (humedad);
  }
  delay(tiempo_espera);
}

void lectura_humedad (void)
{
  digitalWrite(rele1, HIGH);
  delay(2000);
  Serial.print("Humedad de la tierra:");
  Serial.println(analogRead(0));
  digitalWrite(rele1, LOW);
  humedad = analogRead (0);
  Color_Led (humedad);
}

void Color_Led (int humedad )
{

```

```

if (humedad >= 0 and humedad < humedad_min)
{
    analogWrite(redPin, 255);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);
}
if (humedad >= humedad_min and humedad <= humedad_max)
{
    analogWrite(redPin, 0);
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 0);
}
if (humedad > humedad_max)
{
    analogWrite(redPin, 0);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 255);
}
}

void Sensor_Distancia(void)
{
    unsigned int tiempo;
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);          // Calcula la distancia midiendo el tiempo del estado alto
    del pin ECHO
    tiempo = pulseIn(echo, HIGH);    // La velocidad del sonido es de 340m/s o 29
    microsegundos por centimetro
    distancia= tiempo/58;            //manda la distancia al monitor serie
    Serial.print(distancia);
    Serial.println(" cm");
}

void Riego (int )
{
    Sensor_Distancia ();
    if (distancia < distancia_hasta_min); {
        Llenado ();
    }
    lectura_humedad ();              // Realializo una lectura de confirmacion
    if (humedad < humedad_min) {
        digitalWrite(rele1, LOW);
        digitalWrite (rele2, HIGH);
        Serial.println("Regando");
        delay (tiempo_riego);
        digitalWrite (rele2, LOW);
    }
    else {
        Serial.println("No regar");
    }
}

void Llenado (void)
{
    while (distancia < distancia_hasta_max)
    {
        digitalWrite (rele3, HIGH);
        Sensor_Distancia ();
    }
    digitalWrite (rele3, LOW);
}

```

Bibliografía

<http://arduino.cc/es/Tutorial/DigitalPins>
<http://www.instructables.com/id/Fading-RGB-LED-Arduino/3/?lang=es>
[http://www.dfrobot.com/wiki/index.php/Relay_Module_\(Arduino_Compatible\)__\(SKU:_DFR0017\)](http://www.dfrobot.com/wiki/index.php/Relay_Module_(Arduino_Compatible)__(SKU:_DFR0017))
<http://arduino-info.wikispaces.com/Brick-LightSensor-Analog-Digital>
<http://platea.pntic.mec.es/~lmarti2/circuitos/celsolar/comparador.htm>
<http://es.wikipedia.org/wiki/Comparador>
<http://arduino.cc/es/Reference/VariableDeclaration>
<http://arduino.cc/es/Reference/Scope>
<http://www.arduteka.com/foro/>
http://es.wikipedia.org/wiki/Riego_por_goteo
<http://www.hagaloustedmismo.cl/component/hum/proyecto/353/icomohacer-un-riego-por-goteo-y-semiautomatico.html>
http://www.azud.com/Riego/Aplicaciones/riego_por_goteo_subterraneo.aspx
x
Libro Riego por Goteo de Karen Palomino
http://es.wikipedia.org/wiki/Rotaci%C3%B3n_de_cultivos
http://articulos.infojardin.com/huerto/Rotacion_de_cultivos.htm
Libro Buenas Prácticas en Producción Ecológica, Asociaciones y Rotaciones de Gloria I. Guzmán Casado y Antonio M. Alonso Mielgo
(http://www.magrama.gob.es/es/ministerio/servicios/publicaciones/Asociaciones_y_Rotaciones_tcm7-187413.pdf)
<http://www.agrohuerto.com/planificar-el-huerto-la-rotacion-de-cultivos/>
[http://www.dipalme.org/Servicios/Anexos/anexosiea.nsf/VAnexos/IEA-TAE-c4/\\$File/TAE-c4.pdf](http://www.dipalme.org/Servicios/Anexos/anexosiea.nsf/VAnexos/IEA-TAE-c4/$File/TAE-c4.pdf)
<http://www.motril.es/fileadmin/areas/medioambiente/ae/presentacion/documentos/IORotacionCultivos.pdf>
<http://www.monografias.com/trabajos58/riego-goteo-fertirrigacion/riego-goteo-fertirrigacion.shtml>
Libro: Arduino Curso práctico de formación de Óscar Torrente Arteiro editorial RC Libros
Libro: Arduino Práctico de Joan Ribas Iaquera Editorial Anaya.